National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

IDENTIFICATION OF POWER SYSTEM TRANSFER FUNCTIONS USING NEURAL NETWORKS

BY

© DEBORAH MAY GILLARD

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

FALL 1994

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-95036-6

Canada

Name DEBORAH MAY GILLARD

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

ELECTRONICS AND ELECTRICAL
SUBJECT TERM

0544
SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Deborah May Gillard

TITLE OF THESIS: Identification of Power System Transfer Functions Using Neural Networks

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: Fall, 1994

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

SIGNED: D. M. Gillard

ADDRESS: 8725 - 152nd Street
Edmonton, Alberta

DATE: Oct. 4, 1994

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Identification of Power System Transfer Functions Using Neural Networks**, submitted by Deborah May Gillard in partial fulfillment of the requirements for the degree of Master of Science.

K. E. Bollinger (Supervisor)

Q-H. M. Meng

S. L. Shah (External Examiner)

Date: Sept 29, 1994

Abstract

This thesis describes an investigation into the use of a multilayered neural network for measuring the transfer function of a power system. The objectives are to quickly and accurately measure the transfer function of a system with the plant operating under normal conditions. In addition, the excitation signal used in the identification procedure will not affect the quality of the power output or the frequency of the system under test.

This research emphasized the development of a neural network that is easily trained and robust to changing system conditions. Performance studies of the trained neural network are described. In addition, variations in the neural network architecture are also investigated for the purpose of increasing the speed of identification without compromising the accuracy of the results.

Simulation studies suggest the practical feasibility of the algorithm as a stand-alone identification package, even though the algorithm can form part of a feedback control strategy. Finally, the same technique applied to a forward modelling scheme could be used to test the effect of control strategies.

Acknowledgement

The author wishes to thank her supervisor, Professor K. E. Bollinger, for the knowledge, guidance and encouragement given throughout this research.

A special mention to the author's parents, Bob and Joan Gillard, for their never-ending love and faith, and to Linda Meerveld for her hope and vision.

Contents

1 Introduction **1**

1.1 Identification Methods Applied to Power Systems 2

1.2 Background on Neural Networks 5

1.3 Outline of Thesis 12

2 Review of Pertinent Theory **14**

2.1 Training of Neural Networks 15

2.1.1 Static Multilayer Feedforward Neural Network 15

2.1.2 Generalized Delta Rule and Error Backpropagation 18

2.2 Identification as Part of Neural Network Controller 27

3 System Identification of the Two-Machine-Infinite-Bus System **35**

3.1 Frequency Response Model 37

3.2 Off-line Time-Domain Identification 41

3.2.1 Inverse Feedforward Model 42

4 Performance Studies of the Trained Neural Network **58**

4.1 Changes in Operating Conditions 59

4.1.1 Third Order System 60

4.1.2 Decrease Damping and Reduce Frequency of Local Mode 61

4.1.3 Decrease Damping and Increase Frequency of Local Mode 62

4.2 Changes to Neural Network Configuration 68

4.2.1 Feedforward Training 68

4.2.2 Feedback Training 69

4.2.3 Number of Nodes and Node Interconnections 70

4.3 Test on Single-Machine-Infinite-Bus Model 88

5 Summary and Conclusions **90**

5.1 Recommendations for Further Research 94

Bibliography **95**

List of Figures

2.1 Structure of individual node 16

2.2 Three-layered neural network 18

2.3 Logistic sigmoid function 20

2.4 Neural network control configuration 29

2.5 General learning architecture 30

2.6 Indirect learning architecture 31

2.7 Specialized learning architecture 32

2.8 Plant and neural network controller 34

3.1 Block diagram of two-machine-infinite-bus system 36

3.2 Spectrum of PRBS excitation signal 39

3.3 Frequency response of fifth order system 40

3.4 Feedback identification model 43

3.5 Detailed block diagram of feedback identification model . . . 44

3.6 Feedforward identification model 45

3.7 Detailed block diagram of feedforward identification model 46

3.8 Neural network architecture 48

3.9 Error vs. iteration number 53

3.10 Error vs. iteration number showing local minimum 53

3.11 Comparison of time-domain signals after eight training iterations 54

3.12 Comparison of time-domain signals after 10,000 training iterations 55

3.13 Comparison of frequency response after eight training iterations 56

3.14 Comparison of frequency response after 10,000 training iterations 57

4.1 Comparison of frequency response of inverse neural network and third order test system 64

4.2 Comparison of frequency response of inverse neural network and fifth order test system with decreased damping and frequency 65

4.3 Comparison of frequency response of inverse neural network and fifth order test system with decreased damping and increased frequency 66

4.4 Uncoupling of input and output terms 73

4.5 Reduced number of nodes and uncoupling of input and output terms 76

4.6 Test 1: feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 77

4.7 Test 1: feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 77

4.8 Test 1: feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 78

4.9 Test 1: feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 78

4.10 Test 2: feedback training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 79

4.11 Test 2: feedback training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 79

4.12 Test 2: feedback training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 80

4.13 Test 2: feedback training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 80

4.14 Test 3: node reduction - feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 . 81

4.15 Test 3: node reduction - feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 . 81

4.16 Test 3: node reduction - feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 . 82

4.17 Test 3: node reduction - feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 . 82

4.18 Test 4: uncoupling - feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 . 83

4.19 Test 4: uncoupling - feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 . 83

4.20 Test 4: uncoupling - feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 84

4.21 Test 4: uncoupling - feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 84

4.22 Test 5: node reduction and uncoupling - feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 85

4.23 Test 5: node reduction and uncoupling - feedforward training feedforward testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 85

4.24 Test 5: node reduction and uncoupling - feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = 0 to +1 86

4.25 Test 5: node reduction and uncoupling - feedforward training feedback testing, comparison of frequency response of inverse neural network and fifth order test system - input range = -1 to +1 86

4.26 Comparison of frequency response of inverse neural network and single-machine-infinite-bus system 89

List of Tables

4.1 Least squares time-domain error for sinusoidal input 67

4.2 Summary of changes to neural network configuration for feedforward testing 87

Chapter 1

Introduction

Identification theory is a vast subject of considerable interest to the power industry. Due to increasingly complex power systems, the need to quickly and accurately identify the important system characteristics is of prime importance.

In order to apply any type of control action to a power system, whether it be adding controllers or stabilizers, or tuning existing elements, an accurate model of the system is necessary. The model may be obtained either analytically where the model is derived from basic equations and the parameters are assumed, or experimentally using direct measurements.

The following section presents an overview of some of the various methods used in the identification of power systems. This is followed by a review of relevant work utilizing neural networks.

1.1 Identification Methods Applied to Power Systems

A wide range of methods exist for determining the parameters of a synchronous machine. Some test methods require that the machine be isolated from the rest of the system [1,2,4,5,13,38], however the focus here is on identification while the machine is under normal operating conditions.

The Least Squares and Generalized Least Squares time-domain identification techniques were applied to a two-machine-infinite-bus system by Bollinger and Norum [9]. Using Pseudo-Random Binary Sequence (PRBS) excitation and Fast Fourier Transform (FFT) analysis, the frequency responses of the models were compared. The Least Squares identification technique was unable to identify the interarea mode even with pre-filtering of the excitation signal. The Generalized Least Squares technique was able to identify both oscillatory modes but only with the addition of pre-filtering.

On-line frequency response tests were done on a 555 MVA unit by Manchur et al [27]. The turbogenerator was tested over the frequency range of 0.005 to 10 Hz using small sinusoidal perturbations as the exciting input signal. The transfer function was derived from the frequency response, although the method used was not reported. Good correlation between the derived model and the field tests was also

reported.

The Least Squares method of identification was used by Le and Wilson [25] to determine the equivalent circuit parameters of a multi-machine-infinite-bus system. Data simulated using a transient stability program was used to determine the parameters. Good agreement between the model and actual responses was reported, however the addition of random noise showed increased deviation, which indicated an ill-conditioned model under noisy measurement conditions.

Two techniques for deriving transfer functions for automatic voltage regulators (AVR) of power system generators were presented by Bollinger et al [7]. The PRBS excitation and FFT technique were used to obtain frequency response information in the presence of noise. The second technique utilized a Least Squares Parameter Estimation scheme to identify the transfer function parameters. Reasonably accurate parameters were obtained even when initial estimates of 5-10 times the true value were used.

The Least Squares Parameter Estimation scheme of [7] was extended to allow on-line identification in the presence of noise by Bollinger et al in [6]. Comparisons with and without a random disturbance were done on both a second order system and a seventh order system. Good convergence properties were reported and

additional on-site tests produced positive results.

Pseudo-random ternary noise injection was used by Lang et al [23] as an on-line identification technique. Testing was done on a single-machine-infinite-bus system and on an open-circuited machine with an AVR. The system transfer functions were estimated from the results of an impulse response. The effects of random noise were minimized using crosscorrelation techniques. Experimental and theoretical parameters were compared showing reasonable accuracy.

The Generalized Least Squares method of identification was applied to a two-machine-infinite-bus system by Norum [33]. A PRBS input signal with FFT analysis was used to obtain the frequency response model. Comparisons were made using Batch Least Squares identification on the same test system. Improvements for keeping the parameter identification algorithm alert were developed and tested. Implementation under field test conditions were encouraging.

On-line frequency response tests were done on a 60 MW hydroelectric generator by Bollinger et al [10]. The excitation signal was a PRBS with FFT analysis used to obtain the frequency response. Benefits of the PRBS approach over the discrete sinusoidal injection technique were discussed. The system transfer function was derived by making straight-line approximations from the frequency response data.

Time domain averaging was used to demonstrate the noise suppression capabilities of the method.

Transfer function identification in power systems was presented by Smith et al [37]. The paper was a tutorial presentation of identification methods using least squares approaches. An overview of the auto-regressive moving average model and an extension of the Prony signal identification method were discussed. Application examples were given along with the advantages and disadvantages of each method.

1.2 Background on Neural Networks

Considerable work has been done in the past decade using neural networks for the identification and control of power systems. Many different types of neural network architectures and models exist [20,21], however the focus of this research is on static multilayered neural networks.

Much of the existing work done using neural networks is their application as controllers or power system stabilizers on synchronous generators at power plants. Inherent in its use as a controller or stabilizer, is the need for the neural network to first identify the system. In order to achieve power system identification and control, different

researchers have investigated various neural network architectures and training configurations. One such training configuration is the inverse model where the neural network is trained to identify the inverse of the power system. Once the neural network is trained as the inverse of the power system, it can then be put in the forward path so that the gain of the neural network and power system is unity. The plant output can then track a reference input signal.

In transfer function form, a stable system requires all the roots of the denominator polynomial to be located on the left-hand side of the s-plane. In addition, a physically realizable system has a denominator of equal or greater order than the numerator polynomial. A number of problems can result when a model identifies the inverse of such a system.

For the system with numerator roots located in the right-hand side of the s-plane (nonminimum-phase), the inverse would need to have unstable poles. Although the transfer function investigated in this thesis does not have this characteristic, in general the system transfer function is unknown. Widrow and Bilello [41] addressed the nonminimum-phase problem with the use of delayed inverse identification. This involved the use of an adaptive FIR filter with appropriate delay.

Regardless of the numerator term, the denominator polynomial can also cause problems. Since the denominator of the system is of

equal or greater order than the numerator, the inverse will have a numerator term of equal or greater order than the denominator. This produces an infinite amount of derivative action. For this reason, strict pole/zero cancellation, one-step-ahead control, and inverse modelling schemes are not generally proposed outside of academic papers. In practice, these methods are modified by adding a number of denominator terms to the inverse model so that the denominator polynomial is of greater order than the numerator. By carefully choosing the placement of these additional poles, the control action will not be adversely compromised while the negative effects of noise due to derivative action will be filtered out.

An additional consideration in the use of neural networks for identification and control is the use of some form of backpropagation algorithm during training. Since backpropagation is a slow method of training, it is usually used as an off-line procedure. Although in theory the trained neural network can either continue or restart training once operating conditions change, the inherent slowness in updating the neural network weights makes on-line applications more difficult. This difficulty results in research focused on neural networks that are adaptive only during the training phase and have fixed weights during operation. A review of some of the existing strategies is discussed

below.

A detailed analysis of the use of neural networks for the identification and control of dynamical systems was given by Narendra and Parthasarathy [31]. Theory on both static and dynamic backpropagation methods was presented. A number of different plant models that utilize interconnected multilayer and recurrent networks are simulated with promising results regarding the practical feasibility of the schemes.

A neural network based speed control system for a dc motor was presented by Weerasooriya and El-Sharkawi in [39]. Two different identification schemes were investigated with each dependent on the availability of certain motor parameters. Trial and error was used to select all training parameters except the number of neural network inputs and outputs which were problem dependent. The neural network was trained to identify the inverse dynamics of the motor with least-squared error used as the training stopping criteria. Testing was done in the time-domain using tracking of an input reference signal as a measure of successful performance.

An extension of this work was done by Weerasooriya and El-Sharkawi in [40]. The focus of this paper was on the laboratory implementation of a neural network controller for a dc motor. A second

order system was modelled using the same scheme as [39] with training parameters determined by experience. Cross-validation techniques were used as the training stopping criteria and the ability of the neural network to track various reference signals was the performance measure.

A different method of training a neural network power system stabilizer was presented by Zhang et al [42]. A fourth order discrete time input signal was used for the neural network with a single output signal resulting. The neural network was trained to learn the control strategy of an adaptive power system stabilizer (APSS). The APSS was connected to a generator unit which in turn, was connected to an infinite bus. By operating the generator unit under varying conditions, the neural network was trained to respond as the APSS. Time domain tests under varying disturbances were done with comparisons made between the responses of the neural network, the APSS, a conventional stabilizer, and the system without a stabilizer. The neural network provided some advantage over the APSS due to computational considerations, and considerable advantage over the conventional stabilizer and no stabilizer conditions.

Generalized and Specialized architectures for neural network inverse modelling were studied by Psaltis et al [34]. It is the authors' contention that the general learning architecture (referred to as Series-

Parallel in [31]) may result in overtraining since it requires training over a large operational range for generalization. Specialized learning is a method of propagating the error through the plant in order to train the neural network over the more restricted operational range of the plant. Although initial simulations using a combination of both methods failed to provide consistent evidence of improvement, results did point to further research.

Characteristics of error surfaces for multilayer neural networks were presented by Hush et al [22]. The complexity of error surfaces composed of numerous flat and steep regions contribute to the slowness of backpropagation learning. Algorithms with fixed learning rates require small steps so as to avoid oscillations. This contributes to slow learning on relatively flat plateaus. Examples were investigated by simulation. Results also suggested the initialization of weights to small random values about the origin.

A model-reference inverse control system was presented by Widrow and Bilello in [41]. Theory for both forward and inverse identification methods was given. Attention was focused on obtaining a stable inverse for a nonminimum-phase plant. Nonlinear inverse control was achieved by first training the neural network to behave as a forward model of the plant, and then training a separate neural network as an

inverse controller. A nonlinear plant was simulated with the objective of testing the neural network controller's ability to track a command signal. Although the time-domain plots showed good agreement between the command input and the plant output, detailed design parameters and error values were not given.

The layered feedforward neural network was used for synchronous machine modelling by Chow and Thomas [12]. A single-machine-infinite-bus system with known parameters was used for simulation purposes. Two different neural network structures were investigated: one with 10 hidden nodes, and the other with 20 hidden nodes. Each neural network was trained with comparisons between the two methods done using a step response test. The authors conclude that extra hidden nodes increase the model's accuracy. No details regarding neural network parameters, training iterations or performance error were given.

Neural network identification and control of a nonlinear dc brushless motor was investigated by El-Sharkawi et al in [15]. A modified neural network structure was used for the inverse identification of the motor that utilized a speed error term as an input to the neural network and a voltage error term as one of the neural network outputs. Model reference control was then used to test the effectiveness of the trained neural network as a trajectory tracking controller. Comparisons

were made between a PID controller and the neural network for speed tracking, with the neural network providing excellent results. A description of the laboratory setup was included.

1.3 Outline of Thesis

This thesis presents the results of research investigating the development and use of a neural network for system identification with application to a two-machine-infinite-bus system. An inverse model is investigated for implementation and practical use as an identification tool. Performance studies were simulated using both changes in operating conditions and changes in neural network configuration in order to improve the neural network's convergence properties.

A review of the theory pertaining to the training of neural networks is given in Chapter 2. The architecture of the multilayer feedforward neural network is described. The theory of the generalized delta rule for minimizing the error by the method of gradient descent is also given. In addition, the use of an inverse neural network as a controller in the forward path of the system is presented, along with the various architectures and control concepts used.

In Chapter 3, the frequency response of the two-machine-infinite-

bus system used in the simulation is given. The results of the inverse feedforward model of the system is presented in both the frequency and time domain.

The results of performance studies of the trained neural network are presented in Chapter 4. Variations in operating conditions include changes to the damping and frequency of the local mode and testing the neural network on a third order system. The effect of changes to the neural network configuration is investigated by using a Feedforward model versus a Feedback model, by altering the number of nodes and interconnections between the nodes, and by changing the normalization range of the input. The trained neural network is also tested on a single-machine-infinite-bus model.

Chapter 5 presents conclusions gained from the performance studies, and suggests areas for additional research.

Chapter 2

Review of Pertinent Theory

This chapter presents a review of the theory of a class of neural networks known as static multilayer feedforward neural networks. The first identifying feature of this type of neural network is the feedforward flow of information in operational mode, as opposed to recurrent neural networks which encompass feedback at each node. The second feature is the method used for training. The generalized delta rule which minimizes a cost function by the method of gradient descent is presented, and its implementation in the form of error backpropagation is given. Finally, the use of an inverse neural network as a controller in the forward path of the system is discussed, along with the various architectures and control concepts used.

2.1 Training of Neural Networks

2.1.1 Static Multilayer Feedforward Neural Network

The static multilayer feedforward neural network is a specific architecture of the neural network where three or more layers are used. Each layer is comprised of any number of nodes or neurons. The terms static and feedforward refer to the fact that the output of each node is not dependent on past or future outputs, and thus information is only propagated through the network in a forward direction once the neural network has completed training and is in the recall stage.

The number of hidden layers and the number of nodes per layer are not definitive. Although some interesting research and suggestions exist on this problem [14,20,21], it is generally accepted that each specific problem is unique and the configuration must be determined by experience. The number of nodes in both the input layer and the output layer are determined by the system to be identified.

After passing through the input layer, each signal is multiplied by its respective connection weight while fanning out to each node in the next layer. At each node in the next layer, all incoming signals are summed together, along with a bias for that particular node. The output of each node is a nonlinear function, typically a sigmoid or hyperbolic

tangent. The structure of an individual node is shown in figure 2.1.

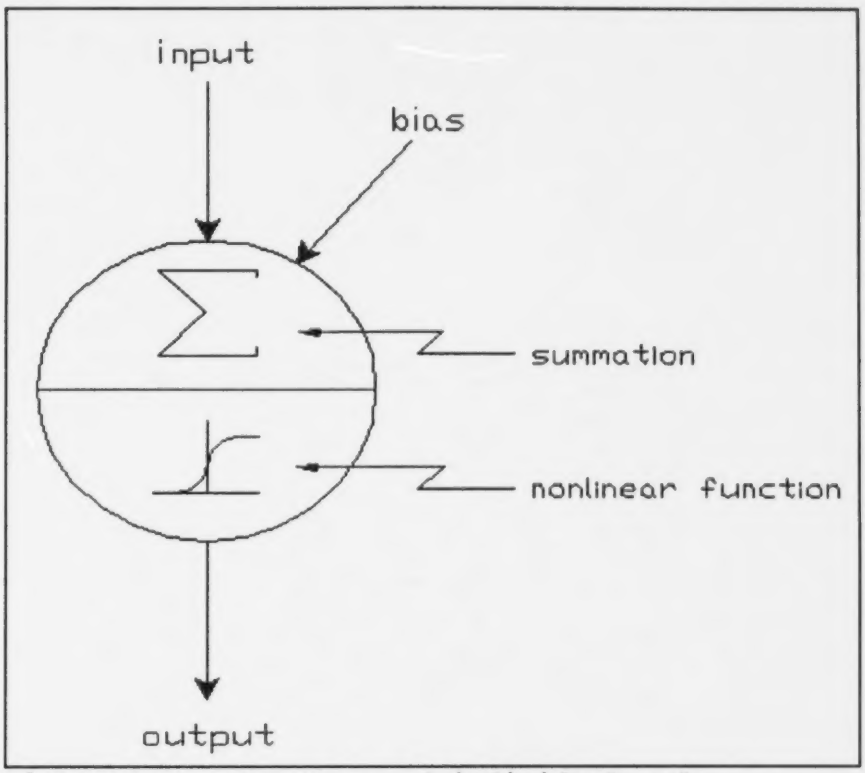

Figure 2.1. Structure of individual node.

For the three-layered neural network shown in figure 2.2, given an input signal, x, the output, y, of each node in the first layer can be expressed as

$$y_i = x_i \tag{2.1}$$

since the first layer is linear. The input to each node of the hidden layer

is then

$$x_j = \sum_i w_{ji}\, y_i + b_j \qquad (2.2)$$

where,

w_{ji} is the weight connection between the j^{th} node of the hidden layer and the i^{th} node of the input layer, and,

b_j is the bias of the j^{th} node of the hidden layer.

The output of each node in the hidden layer is passed through a nonlinear function. Using the logistic sigmoid results in the output,

$$y_j = \frac{1}{1 + \exp(-x_j)} \qquad (2.3)$$

The input to each node of the final layer is then,

$$x_k = \sum_j w_{kj}\, y_j + b_k \qquad (2.4)$$

The output of each node of the last layer is,

$$y_k = \frac{1}{1+ \exp(-x_k)} \tag{2.5}$$

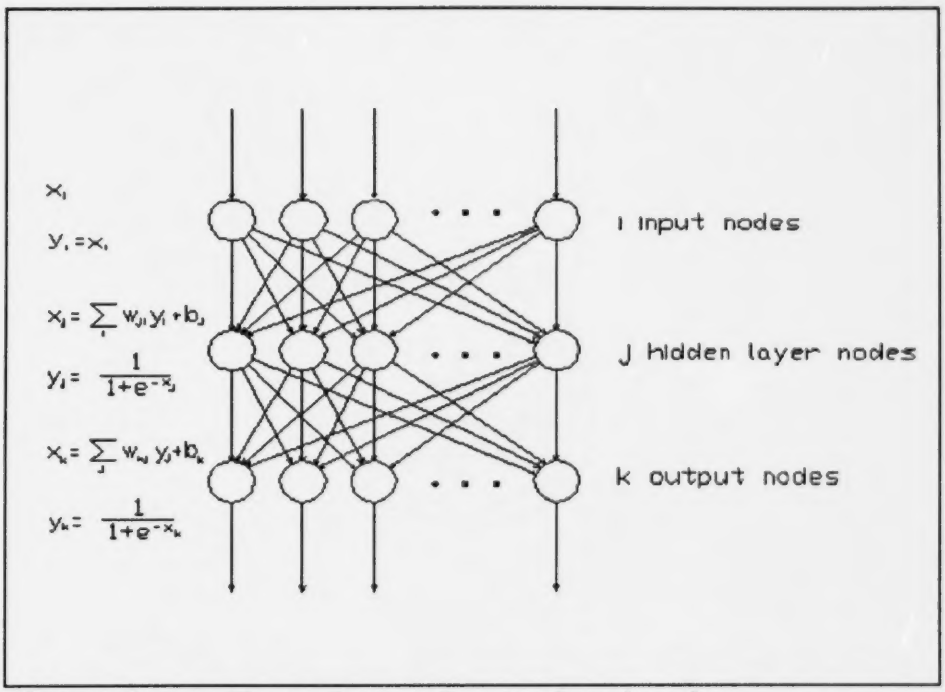

Figure 2.2. Three-layered neural network.

2.1.2 Generalized Delta Rule and Error Backpropagation

The generalized delta rule is one of many least-mean-square methods of learning. The least-mean-square method of learning seeks to minimize a performance error. The generalized delta rule uses this

performance error to find a minimum by gradient descent [35]. Error backpropagation is the specific method of assigning this error for the adjustment of the neural network weights.

Defining the error to be minimized as,

$$E = \sum_p E_p = \frac{1}{2} \sum_p \sum_k (y_d - y_k)^2 \qquad (2.6)$$

where,

y_d is the desired output, and,

y_k is the actual output.

The error is summed over all k output nodes and all p input patterns presented for training.

In order to assign this error to each weight in the network, the gradient is calculated, beginning with the output node, and backpropagated through the various layers. The backpropagation of the error ends at the input layer because the output of the first (linear) layer is a constant which produces a gradient of zero. This is why the gradient descent method necessitates a continuous function as the output of each node. In order to backpropagate the error through each layer, a non-zero gradient is needed in order to update the weights of the network.

The logistic sigmoid function is particularly well-suited for

gradient descent algorithms, because it is a bounded, monotonic, non-decreasing function with a positive derivative. It is described by,

$$f(x) = \frac{1}{1 + \exp^{-\alpha x}} \tag{2.7}$$

where α denotes the steepness of the transition period. The value of α is usually taken to be unity, as seen in figure 2.3, and that value is used throughout this work.

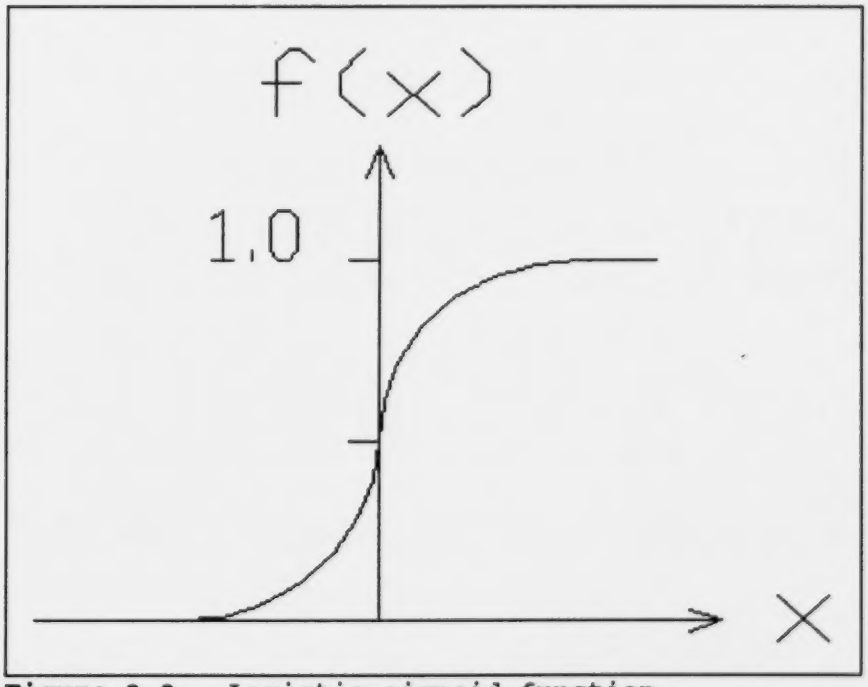

Figure 2.3. Logistic sigmoid function.

Using the chain rule for partial derivatives, the error for the output layer with respect to the output layer's weights can be expressed as,

$$\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial x_k} \frac{\partial x_k}{\partial w_{kj}} \tag{2.8}$$

Taking each partial derivative separately yields,

$$\frac{\partial E}{\partial y_k} = -(y_d - y_k) \tag{2.9}$$

$$\frac{\partial y_k}{\partial x_k} = y_k(1 - y_k) \tag{2.10}$$

$$\frac{\partial x_k}{\partial w_{kj}} = y_j \tag{2.11}$$

This determines the steepest ascent of the gradient, and thus to find the minimum, the required error is the negative of the gradient.

$$-\frac{\partial E}{\partial w_{kj}} = (y_d - y_k)\, y_k\, (1 - y_k)\, y_j \tag{2.12}$$

In the same manner, the error can be assigned to the hidden layer weights using the chain rule of partial derivatives.

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial x_k} \frac{\partial x_k}{\partial y_j} \frac{\partial y_j}{\partial x_j} \frac{\partial x_j}{\partial w_{ji}} \tag{2.13}$$

The resulting negative of the gradient is then given by,

$$-\frac{\partial E}{\partial w_{ji}} = (y_d - y_k)\, y_k\, (1-y_k)\, w_{kj}\, y_j\, (1-y_j)\, y_i \tag{2.14}$$

The change in each weight is then assigned according to,

$$\Delta w_{kj} = \eta \left[-\frac{\partial E}{\partial w_{kj}} \right] \tag{2.15}$$

and,

$$\Delta w_{ji} = \eta \left[-\frac{\partial E}{\partial w_{ji}} \right] \tag{2.16}$$

where η is a constant that determines the step size. There are no

definitive rules for determining the value of η. The greater the step size, the faster the network can theoretically reach a minimum, however too high a number can lead to oscillations while too low a number can result in slow convergence.

Some research exists proposing adaptive learning rates [11,19] but caution that results may be problem dependent.

The change in each bias is also assigned according to the same procedure. For the bias in the output layer, the chain rule of partial derivatives gives,

$$\frac{\partial E}{\partial b_k} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial x_k}\frac{\partial x_k}{\partial b_k} \tag{2.17}$$

which results in a negative gradient given by,

$$-\frac{\partial E}{\partial b_k} = (y_d - y_k)\, y_k\, (1 - y_k) \tag{2.18}$$

In the same manner, the chain rule for the bias in the hidden layer gives,

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial x_k}\frac{\partial x_k}{\partial y_j}\frac{\partial y_j}{\partial x_j}\frac{\partial x_j}{\partial b_j} \tag{2.19}$$

which results in a negative gradient as follows,

$$-\frac{\partial E}{\partial b_k} = (y_d - y_k)\, y_k\, (1-y_k)\, w_{kj}\, y_j\, (1-y_j) \tag{2.20}$$

The change in each weight is assigned by the following rule,

$$\Delta b_k = \eta \left[-\frac{\partial E}{\partial b_k} \right] \tag{2.21}$$

and,

$$\Delta b_j = \eta \left[-\frac{\partial E}{\partial b_j} \right] \tag{2.22}$$

Once an input is presented, the signal is propagated through the network until an output is generated. This actual output is compared to the desired output, and the required gradients are calculated. After all of the input patterns have been presented, one training iteration is

complete. The weights are then adjusted using the sum of all the weight changes over one training iteration. This can be expressed as,

$$w_{kj}(m+1) = \sum_{p} \Delta w_{kj} + w_{kj}(m) \tag{2.23}$$

and,

$$w_{ji}(m+1) = \sum_{p} \Delta w_{ji} + w_{ji}(m) \tag{2.24}$$

where m is the iteration number.

Similarly, the bias values are adjusted using the sum of all the bias changes over one training iteration and is given by,

$$b_k(m+1) = \sum_{p} \Delta b_k + b_k(m) \tag{2.25}$$

and,

$$b_j(m+1) = \sum_{p} \Delta b_j + b_j(m) \tag{2.26}$$

This is termed learning be epoch [29], however if the learning rate is small, the weights may be adjusted after the presentation of each pattern [29,32].

This forward propagation of signals and backpropagation of error is continued until a stopping criteria is met. For example, a set number of iterations or a threshold error may be used as the training stopping criteria.

In order to speed up learning without causing oscillations, the addition of a momentum term is often used. The momentum term, α, stresses past weight changes in determining current movement. This allows the change in weight calculation to be expressed as,

$$\Delta w_{kj}(m+1) = -\eta \left[\frac{\partial E}{\partial w_{kj}} \right] + \alpha \, \Delta w_{kj}(m) \tag{2.27}$$

Although the same solution may be reached with a smaller step size, McClelland and Rumelhart have found that systems learn quicker by increasing the learning rate and using a momentum term [29]. In [35], Rumelhart et al recommend a learning rate of 0.25 and a momentum term of 0.9 as a starting point.

The weights and biases of the neural network must be set before

training begins. It is generally accepted that low initial weights allow the network to start learning in a relatively safe position, thus ensuring convergence over a wide range of problems. Although high initial weights can accelerate learning, values in the interval [-0.3,0.3] ensure that reliable learning is maintained [24].

2.2 Identification as Part of Neural Network Controller

The motivation for utilizing an inverse neural network for identification purposes, is its subsequent use as a controller in the forward path of the system. Although many different architectures and configurations have been proposed, only a few that are particularly relevant to the identification strategy in this thesis will be discussed in detail below.

A neural network controller based on indirect Model Reference Adaptive Control was presented by Weerasooriya and El-Sharkawi in [40]. Inverse plant identification was accomplished by assuming a functional relationship (f), of the plant as follows,

$$y_p(k) = f\,[u_p(k-1),\ y_p(k-1),\ y_p(k-2)] \tag{2.28}$$

where,

$y_p(k)$ is the plant output, and,

$u_p(k-1)$ is the past plant input.

In order to represent the inverse dynamics of the plant, the output of the neural network can then be expressed in the form of a functional relationship (g),

$$u_n(k-1) = g[y_p(k),\ y_p(k-1),\ y_p(k-2)] \tag{2.29}$$

where,

$u_n(k-1)$ is the neural network output.

The relationship described by equation (2.29) avoids any problems with additional past plant input terms in the functional relationship. An additional past input term, $u_p(k-2)$ in equation (2.28) would require either $u_p(k-2)$ (feedforward model) or $u_n(k-2)$ (feedback model) in equation (2.29). The feedforward and feedback models are discussed further in Chapter 3.

The training input sequence was a sum of sinusoids set to excite the frequencies of interest. Training was stopped by using cross-validation techniques, where the training error was compared with the

error of an independent data set of the same form.

The identification procedure was done off-line, with the resulting fixed weights used in the neural network controller. The control concept is shown in figure 2.4 with on-line identification, however only off-line training was done. Also shown is the addition of a fixed gain feedback loop used to compensate for the effects of dead-band.

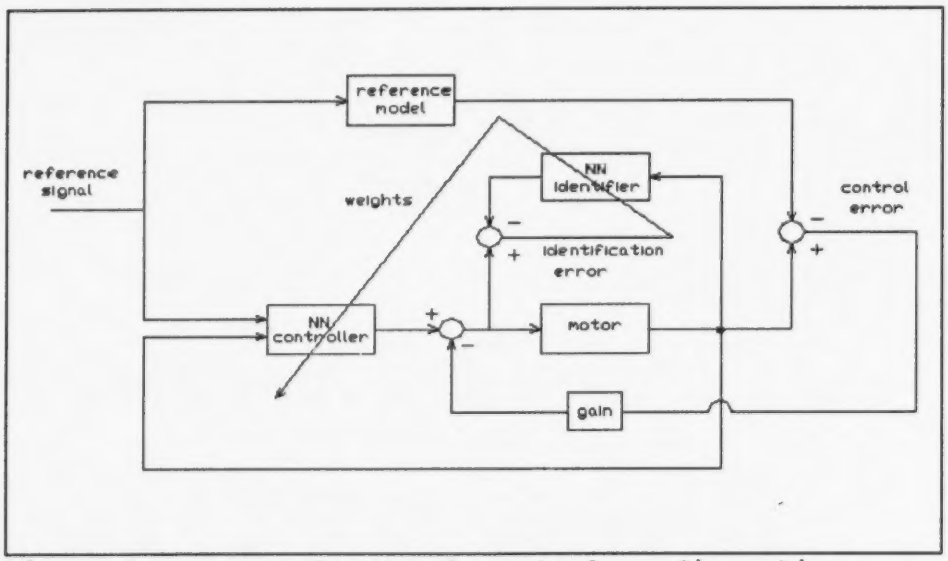

Figure 2.4. Neural network control configuration.

A suitable reference model was chosen which determined the detailed design coefficients and time delays used in the controller, but

not shown in figure 2.4. Controller performance was tested by evaluating the tracking performance of the system and included the effects of noise. The neural network controller performed high speed tracking accurately, however, the steady state error in the position tracking indicated the need for integral gain in the control loop. The authors [40] also felt that better control performance could be achieved with on-line training.

A specialized learning architecture was proposed for on-line training of a neural network controller by Psaltis et al in [34]. A discussion of the disadvantages of other training architectures preceded the proposed method. The authors [34] refer to the typical procedure of off-line learning as the general learning architecture, as seen in figure 2.5.

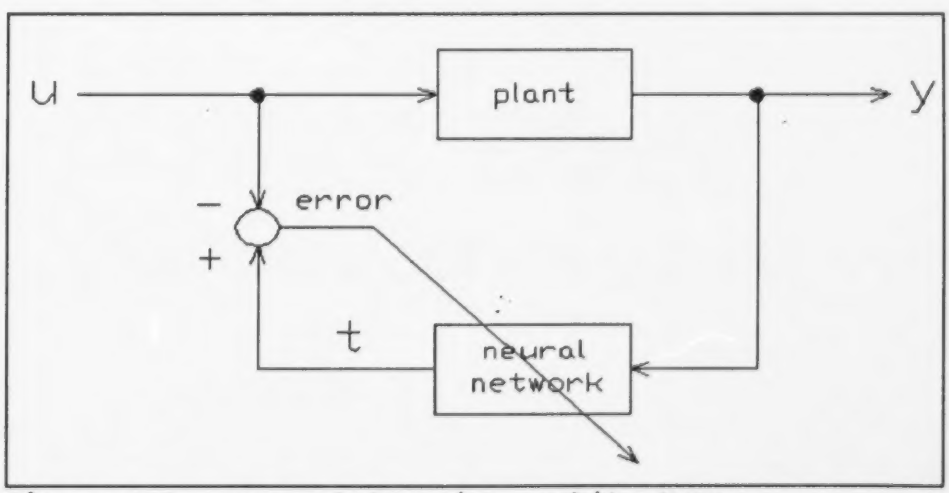

Figure 2.5. General learning architecture.

Success of this method requires training the neural network over a broad operating range to ensure generalization of the neural network to inputs not specifically seen in training. They believe that this method could be inefficient and less accurate than a method where a specific operating range is focused on.

The second method of training discussed was the indirect learning architecture which is shown in figure 2.6. This method does allow training over the specific region of interest since all signals are generated from the desired response.

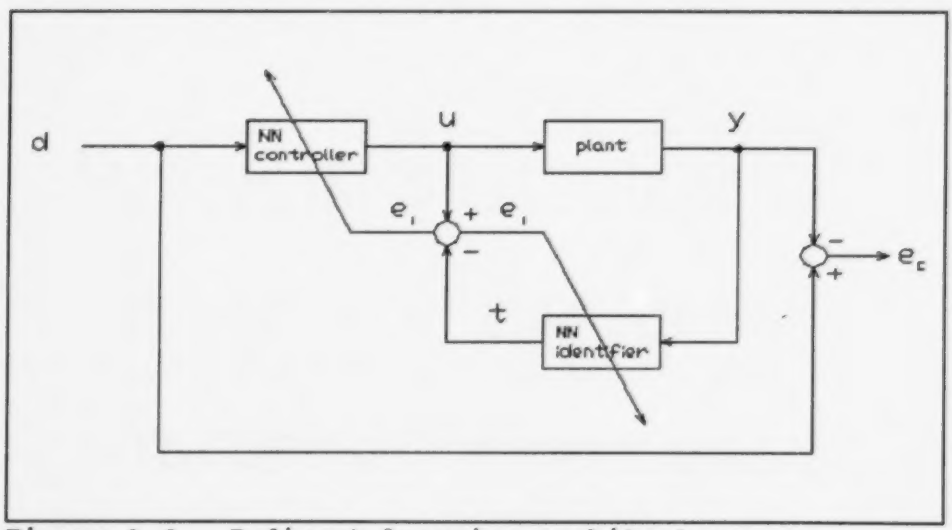

Figure 2.6. Indirect learning architecture.

However, the possibility exists that by mapping all the desired signals d, to a single control signal $u=u_0$, then the output signals y would be mapped to a single neural network control signal $t=u_0$, thus minimizing the identification error e_i, but not minimizing the control error e_c.

Instead, they proposed an on-line method of training that minimized the control error, as seen in figure 2.7.

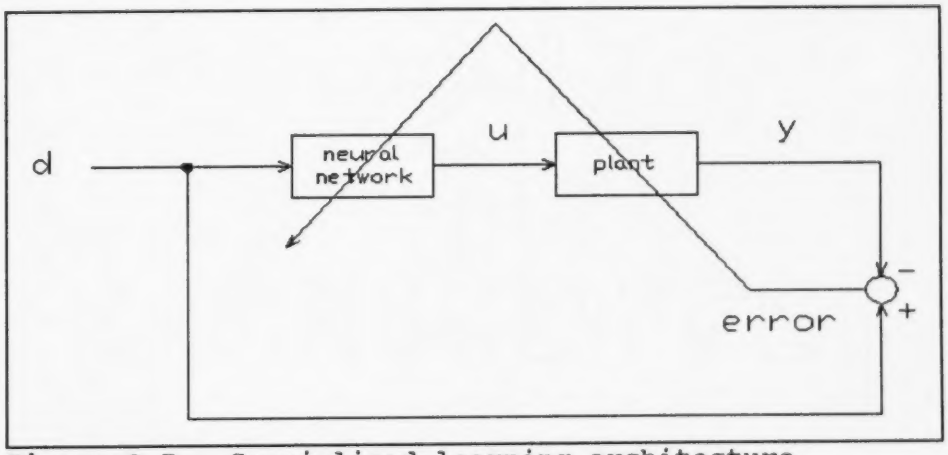

Figure 2.7. Specialized learning architecture.

However, since the plant is now between the error and the neural network, a method of determining the Jacobian of the plant would be necessary. If the plant is unknown, which is usually the case, the partial derivatives of the plant output at a particular operating point can be

estimated, thus allowing the error in the output to be backpropagated to the neural network. This configuration then allowed the neural network to be trained over a specialized region of interest.

The authors [34] believe that a combination of both generalized learning and specialized learning might be best. Initial training over a general region could set the weights to values that facilitate better learning using the specialized learning architecture. Although they were unable to determine what conditions would yield consistently improved performance, results were promising.

Saerens and Soquet [36] tested the specialized learning architecture proposed by Psaltis et al [34] for the entire training procedure, as opposed to a combination of generalized and specialized training. They also proposed different methods of approximating the partial derivatives of the plant. Simulations were done comparing the proposed schemes with standard backpropagation. Both methods performed equally well.

Nguyen and Widrow [32] also proposed a method of training the neural network controller in the forward path of the system, as shown in figure 2.8. Based on optimal control, the neural network controller was trained to produce the required control signal u_n, to drive the plant to the desired state z_d, given the current state of the plant z_k.

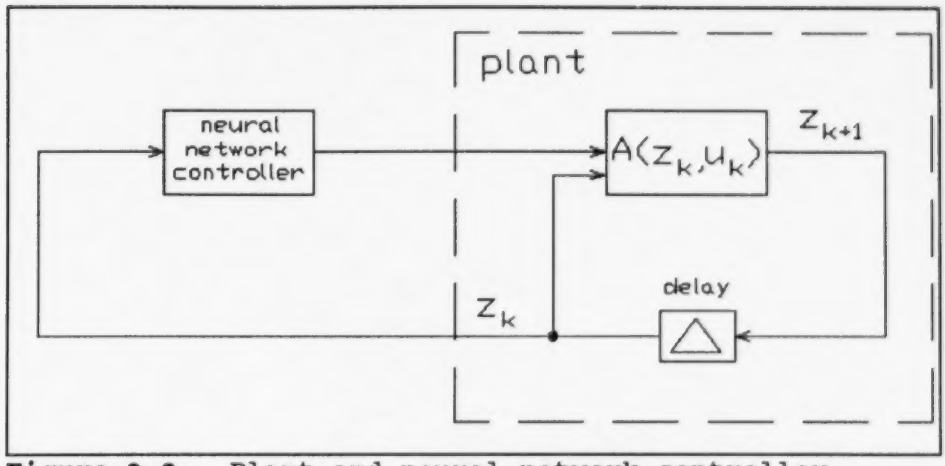

Figure 2.8. Plant and neural network controller.

In order to backpropagate the error at the plant output, they proposed to first train a separate neural network to identify the forward model of the plant. The error between the final state and the desired final state can be backpropagated through the neural network forward model of the plant to the neural network controller with only the weights of the neural network controller being adjusted.

The proposed scheme was simulated by training a neural network controller to steer a trailer truck to back up to a loading dock. Once training was completed, the controller was able to successfully back up the trailer truck from a variety of initial positions.

Chapter 3

System Identification of the Two-Machine-Infinite-Bus System

The ability to adequately represent the important dynamics of a system, over a wide bandwidth, is of paramount importance for all identification schemes. Any control action or stabilizer design for a power system will only be as good as the model it is based on.

The power system used in the simulation studies is very similar to that used by Norum [33]. Parameters were chosen to give characteristics commonly found in the real world. A block diagram of the power system is shown in figure 3.1.

This representation simulates a single generator in the form of the test machine, connected to a large generation/load pool in the form of the equivalent machine. In addition, the infinite bus represents the interconnection to an even larger generation/load pool.

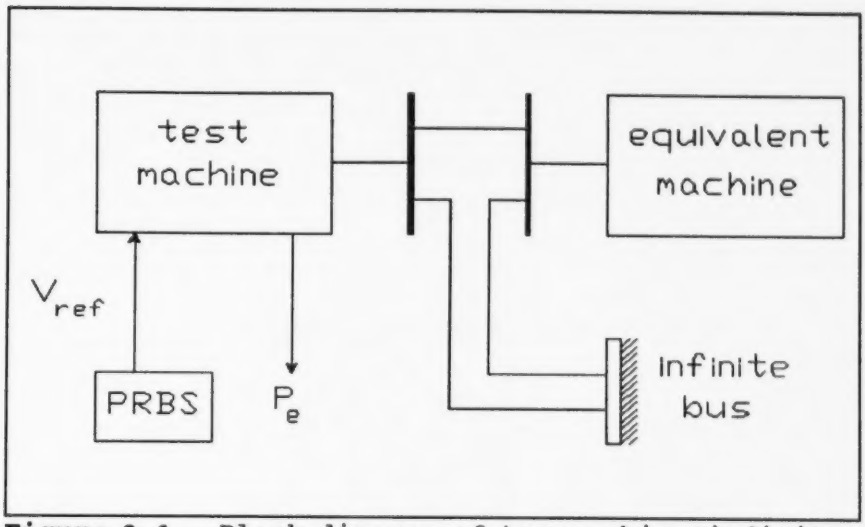

Figure 3.1. Block diagram of two-machine-infinite-bus system.

The system's two oscillatory modes are the local and interarea modes. The local mode of oscillation corresponds to the test machine swinging relative to a large generation pool. The interarea mode of oscillation is that of the large generation/load pool swinging relative to the infinite bus. The damping and frequency terms of the local and interarea mode were selected to give reasonable relative oscillations between them.

The neural network was used to obtain the inverse model of the power system. The accuracy of identification was then tested using frequency response techniques, and the results are presented in the

following section.

3.1 Frequency Response Model

The two-machine-infinite-bus model can be approximated by a fifth order transfer function linking the electric power output (P_e) to its AVR PSS reference voltage input (V_{ref}) . This is given by,

$$\frac{P_e}{V_{ref}} = \frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+144)(s^2+0.1s+9)} \tag{3.1}$$

This transfer function was used as a starting point for testing the various neural network architectures.

As a basis for comparison, the frequency response of (3.1) was simulated using PRBS excitation and FFT analysis [3]. The PRBS signal was composed of 512 points from a random number generator with a minimum clock pulse width of 0.2 seconds. The sampling rate was set to 40 Hz.

The frequency response of (3.1) could also have been calculated using the substitution s=j ω, and then calculating the magnitude and angle at various frequencies over the bandwidth of interest. However, since the frequency response of the neural network was done using PRBS

excitation, it was arbitrarily decided to use the same technique for the frequency response of the power system model.

In order to establish periodicity of the input signal, a quarter period of the PRBS signal was applied as a leading signal to the plant [8]. This allows the initial transient response of the system to decay to approximately zero. Thus, one and a quarter periods of PRBS excitation signal were applied to the power system model, with only the final period of input/output time domain data used in the spectral analysis.

All tests were done without the benefit of a low-pass filter, however, for the purpose of illustration, a 4096 point PRBS excitation signal was put through a sixth-order low-pass butterworth filter with a cut-off frequency of 5 Hz and sampled at 40 Hz. The spectrum of the PRBS excitation input signal and the frequency response of the fifth order transfer function model (P_e / V_{ref}) are shown in figures 3.2 and 3.3 respectively. The spectrum of the PRBS excitation signal shows a fairly uniform energy content over the frequency range of interest, with the first null determined by the clock pulse width.

The frequency response of the fifth order transfer function model clearly shows both the local and interarea modes of oscillation. The interarea mode is evident at 0.48 Hz. as is the local oscillatory mode at 1.9 Hz.

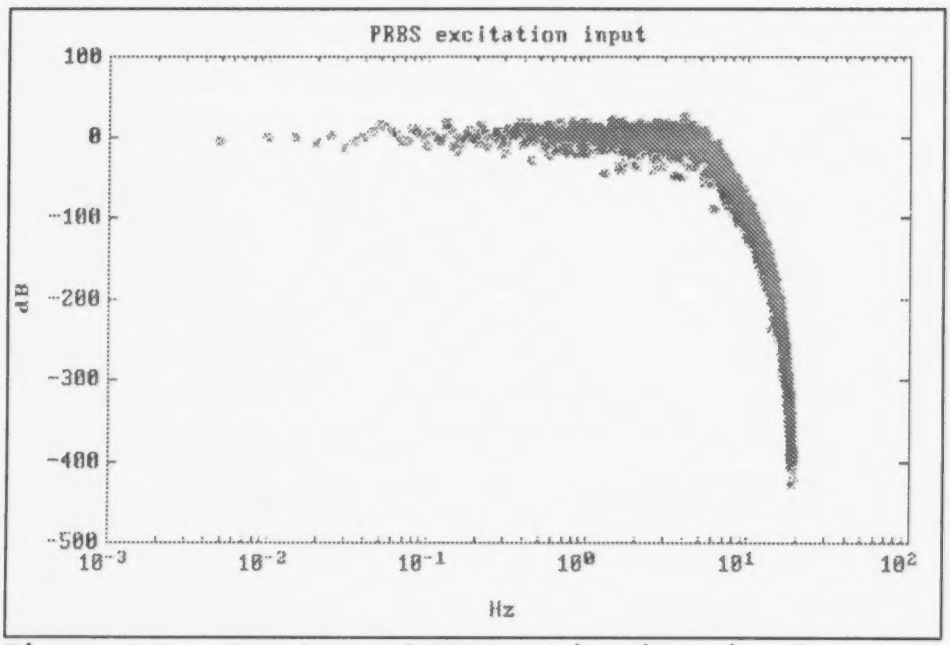

Figure 3.2. Spectrum of PRBS excitation signal.

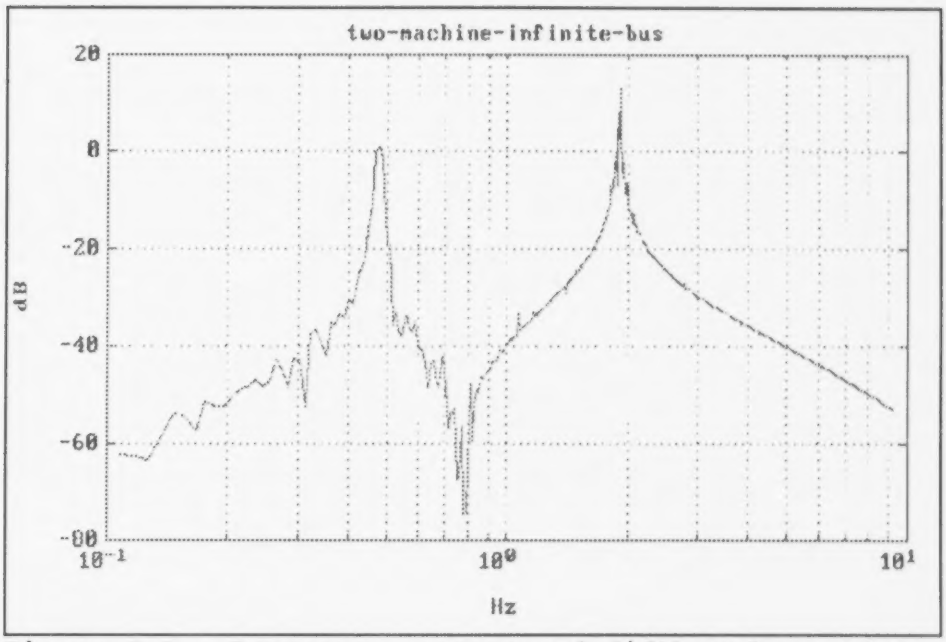

Figure 3.3. Frequency response of fifth order system.

3.2 Off-line Time-Domain Identification

The difficulties associated with the identification of a lightly damped fifth order model, where the local and interarea modes are closely located in the frequency domain but differ by more than 10 dB, are considerable. In [33], Norum tested a similar system and found that neither the Batch Least Squares algorithm nor the Generalized Least Squares technique were able to identify the interarea mode with only a low-pass filtered PRBS as the excitation input. In order to provide more energy in the excitation sequence near the interarea mode, Norum found it necessary to employ both a low-pass filter and a notch-filter for excitation shaping. Additional information on the role of pre-filters is given by Ljung in [26].

The same idea has been employed in this work. Since the simulated test system is known, and because the neural network is set to identify the inverse of the test system, the ultimate pre-filter was employed by putting the PRBS excitation signal through the test system, and then using the test system output as the input signal to the neural network. The test system acts like a pre-filter so that a higher energy level is applied at the oscillatory modes. No other filtering was used during testing.

3.2.1 Inverse Feedforward Model

The base case neural network used was an inverse feedforward model. The fifth order test system of (3.1) is re-written in terms of the power system output y_p , and input u_p , as follows,

$$\frac{y_p(s)}{u_p(s)} = \frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+9)(s^2+0.1s+144)} \tag{3.2}$$

which results in a difference equation of the form

$$y_p(k) = \sum_{i=1}^{5} a_i\, y_p(k-i) + \sum_{j=1}^{5} b_j\, u_p(k-j) \tag{3.3}$$

Since the test system output is a function of five delayed inputs and five delayed outputs, it follows that (3.3) can be rearranged in terms of $u_p(k-1)$ and some function (f). This results in the following relationship,

$$u_p(k-1) = f\begin{bmatrix} u_p(k-2),u_p(k-3),\ldots,u_p(k-5), \\ y_p(k),y_p(k-1),\ldots,y_p(k-5) \end{bmatrix} \tag{3.4}$$

In order for the neural network to identify the inverse dynamics of the test system, the output of the neural network can also be written as a function (g) of its inputs.

$$u_n(k-1) = g\begin{bmatrix} u_n(k-2),u_n(k-3),\ldots,u_n(k-5), \\ y_p(k),y_p(k-1),\ldots,y_p(k-5) \end{bmatrix} \tag{3.5}$$

The Feedback Identification Method can be drawn in general block diagram form as shown in figure 3.4, and in figure 3.5 for the specific case given by equation (3.5).

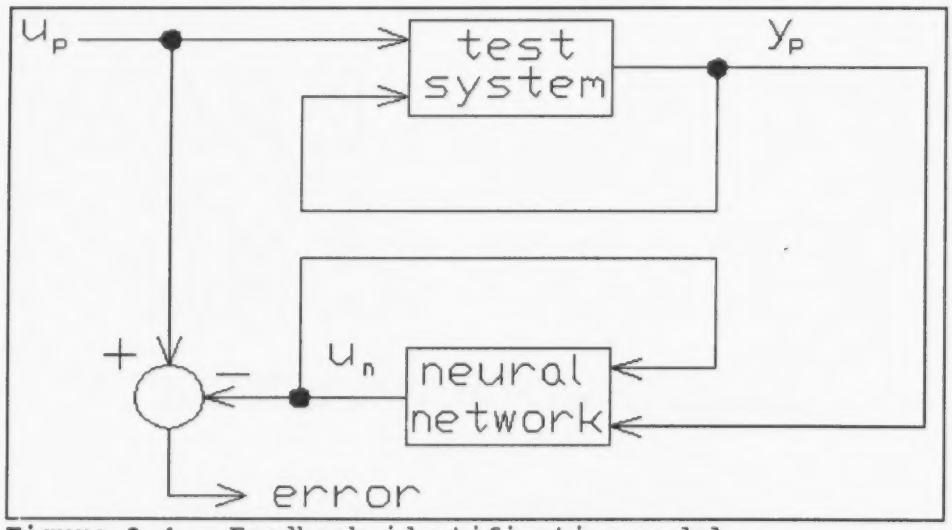

Figure 3.4. Feedback identification model.

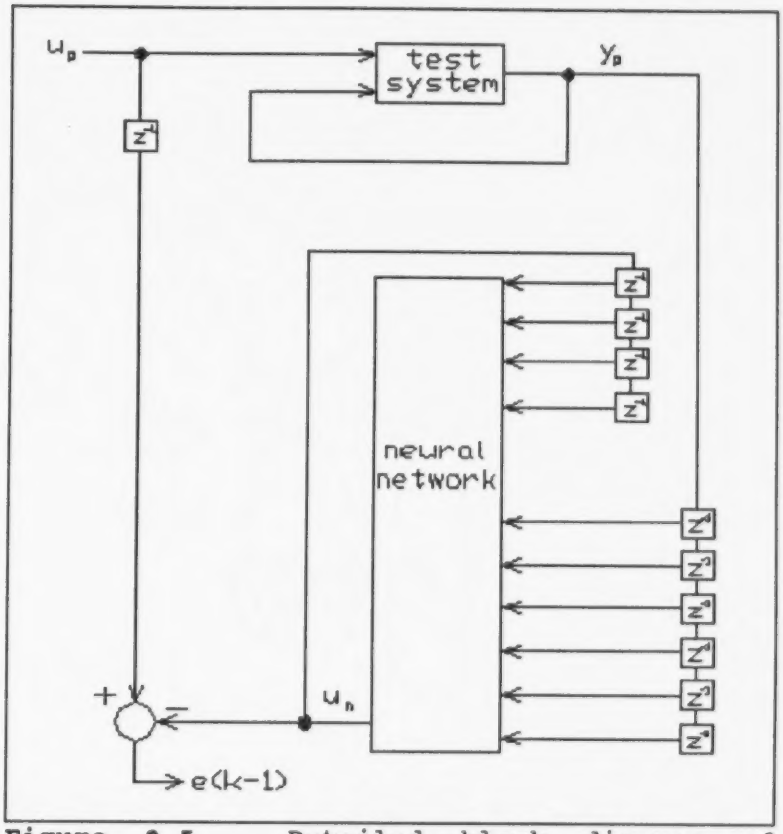

Figure 3.5. Detailed block diagram of feedback identification model.

However, although the test system can be assumed to be bounded-input bounded-output stable, the neural network cannot. In fact, no theory currently exists which can guarantee convergence of the feedback model parameters [31]. This necessitates the use of the Feedforward Identification Method, which is shown for the general case in figure 3.6,

where the power system input is also fed as an input into the neural network instead of using the neural network output as a feedback term. This results in the following relationship,

$$u_n(k-1) = g\begin{bmatrix} u_p(k-2),u_p(k-3),\ldots,u_p(k-5), \\ y_p(k),y_p(k-1),\ldots,y_p(k-5) \end{bmatrix} \tag{3.6}$$

Thus, all the input signals used in the identification process are bounded. The block diagram specific to (3.6) is shown in figure 3.7.

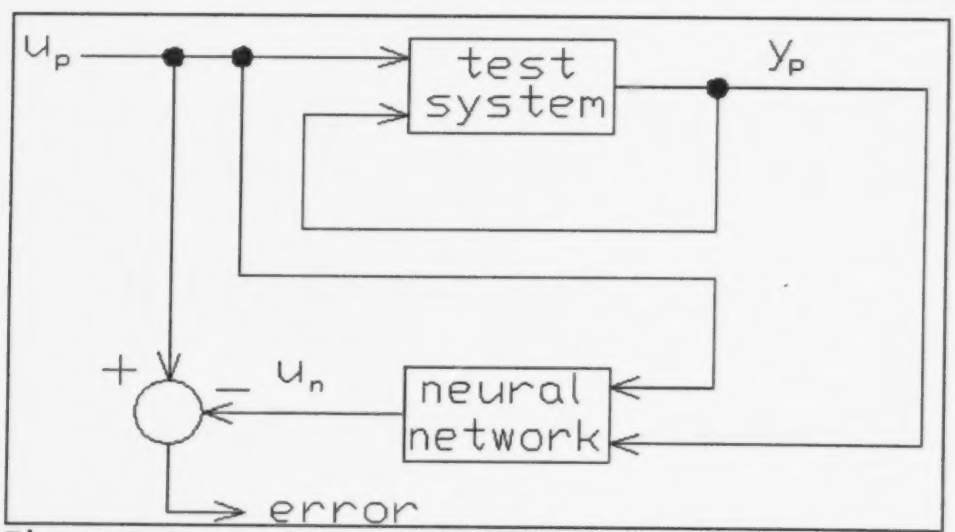

Figure 3.6. Feedforward identification model.

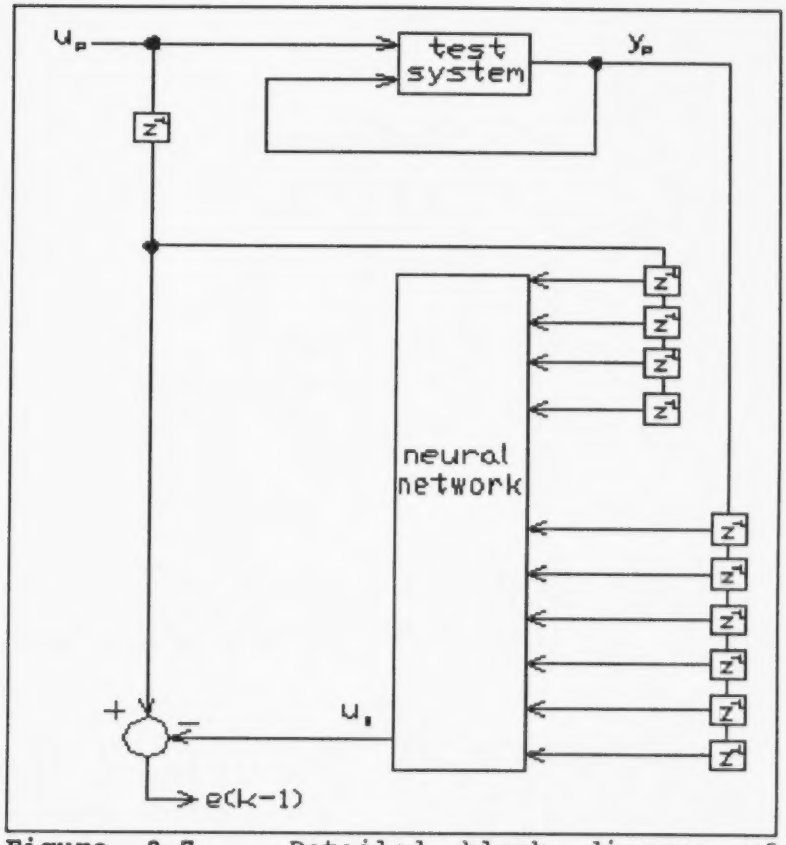

Figure 3.7. Detailed block diagram of feedforward identification model.

Theoretically, once the neural network output, u_n ,approaches the power system input, u_p , (the error tends to a small value asymptotically), the feedforward model can be replaced by the feedback model.

From the functional relationship form of the inverse model given in (3.6), it can be seen that in order to generate the single output $u_n(k-1)$, the neural network must have one node in the output layer. From the same equation, it is also necessary to have ten inputs to the input layer of the neural network. This representation accounts for all four delayed power system inputs, the one present power system output, and all five past power system outputs. It was decided to have the neural network configuration with one hidden layer and twenty hidden layer nodes. The number of hidden layer nodes was set by comparing the sum of the squared error for eight training iterations using ten, twenty, and thirty hidden layer nodes. Additional consideration was given to the fact that added nodes would increase computational time, since more weights would need to be updated at each iteration. Twenty hidden layer nodes produced the lowest error while maintaining a reasonable iteration time. The neural network architecture can be seen in figure 3.8.

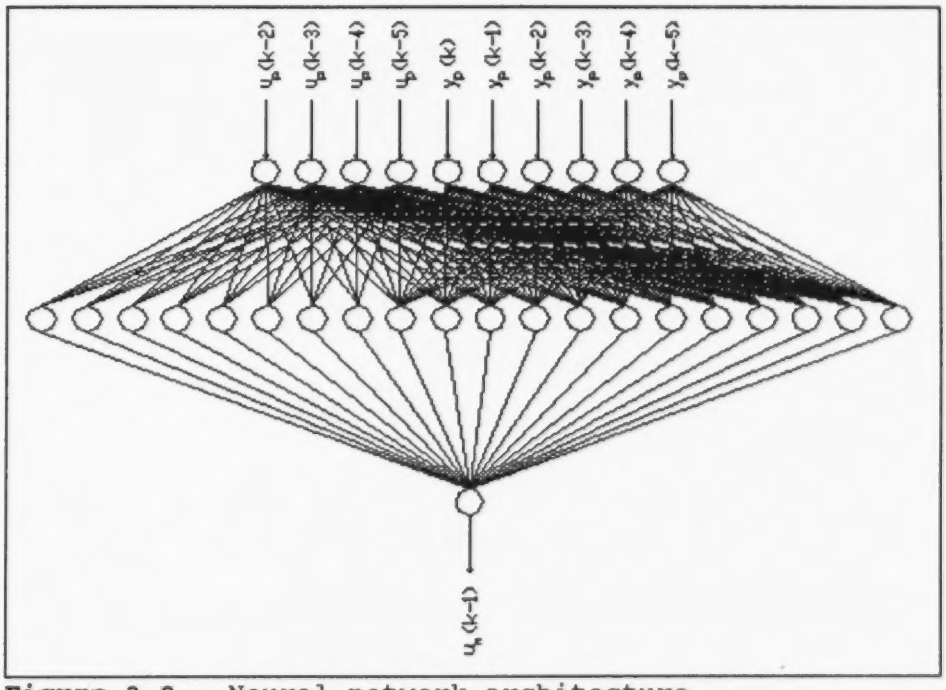

Figure 3.8. Neural network architecture.

The training input signal was of the same form used by Weerasooriya and El-Sharkawi in [40] and is given by,

$$u_p(k) = 500 \sin\left[\frac{2\pi kT}{\alpha}\right] + 450 \sin\left[\frac{2\pi kT}{\beta}\right] + 250\left[1 - \exp\left(\frac{-kT}{.03}\right)\right] \tag{3.7}$$

The sampling time was set to 30 ms and 600 points were generated. The parameters α and ß are chosen here to be 2.1 and 0.52 respectively, so as to excite the local and interarea oscillatory modes. The sinusoidal form of the training input is set to give the neural network a broad range of operating conditions.

Since the nonlinear function chosen is the logistic sigmoid of (2.7), the inputs must be normalized between zero and one (the output range). Therefore, the data file containing the test system input and output are normalized by using the maximum and minimum values as shown below,

$$u_p(norm) = \frac{u_p - \min(u_p)}{\max(u_p) - \min(u_p)} \tag{3.8}$$

and,

$$y_p(norm) = \frac{y_p - \min(y_p)}{\max(y_p) - \min(y_p)} \tag{3.9}$$

The values for the learning rate and momentum term are chosen to be 0.25 and 0.9 respectively.

The weights and biases of the neural network are chosen using a random number generator and are set in the interval [-0.1,0.1] so that none of the nodes start too close to saturation.

The neural network is then trained in the inverse feedforward model configuration with the weights updated after the presentation of each pattern.

The base case test system was allowed to train for a number of iterations to see what the error function looked like relative to the iteration number. Typically, research that utilized neural networks for the identification and control of power systems have focused on the control aspect. Thus the identification procedure consisted of thousands of iterations to minimize the time-domain error. Anywhere from 1000-500,000 iterations are generally used [15,17,18,28,30,31,32,34,39,40].

Referring to figure 3.9, the sum of the squared error appears to decrease gradually throughout the training. A closer inspection of the training error reveals a local minimum at the eighth iteration, as seen in figure 3.10.

The time-domain output signal of the neural network is plotted against the desired output signal at the eighth iteration and is shown in

figure 3.11. It can be seen that although the frequency of the two signals is the same, there is a considerable difference between the neural network output and the desired output. The least squares error summed over all 600 input patterns is calculated for the test system using the error function defined by equation (2.6). For the sinusoidal training input of (3.7), the resulting least squares error is 0.406 for the weights that were frozen at the eighth iteration.

The same comparison is again made after 10,000 training iterations. Referring to figure 3.12, the difference between the output of the neural network and the desired output is indiscernible. The sum of the squared error has now decreased to 0.00459 over all 600 points.

An examination of the frequency responses of the neural network and the fifth order system was done after eight iterations and after 10,000 iterations. Since the neural network is set to identify the inverse of the test system, all frequency responses were done on the inverse of the neural network. The frequency response was done using the PRBS input signal and FFT analysis described in section 3.1 of this thesis. The results of both the test system and the inverse neural network are shown in figures 3.13 and 3.14. There is basically no difference between the frequency response of the neural network when the training is stopped after eight iterations and when the training is stopped after 10,000

iterations. No obvious improvement has resulted from the increased training. Clearly the neural network has captured both oscillatory modes of the test system after only eight iterations. In addition, the plots show that any difference in the magnitude of the peaks is indiscernible. The amount of dB per decade drop-off of the local mode is slightly more for the inverse neural network than the fifth order test system.

Since it is necessary for a viable identifier to converge quickly, while capturing the important characteristics of the plant, it was decided to do all testing using eight iterations as the benchmark for this system. It is at this point that weights are frozen for testing the relative convergence to the known model of the plant.

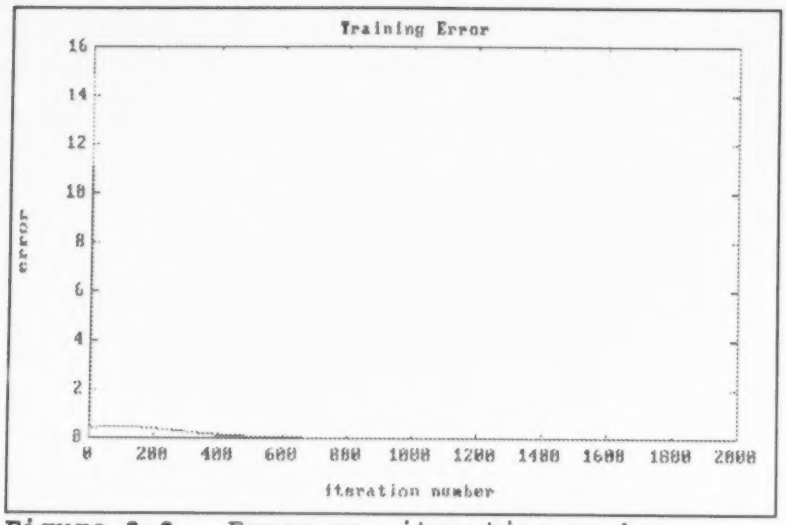

Figure 3.9. Error vs. iteration number.

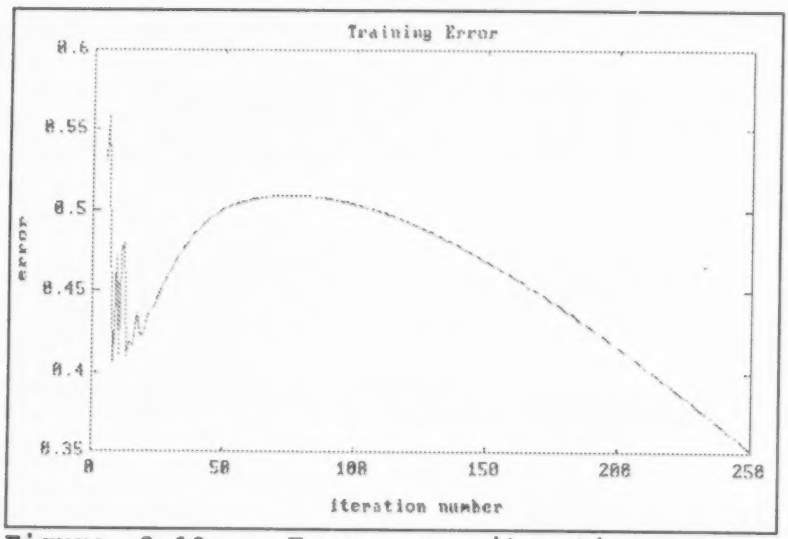

Figure 3.10. Error vs. iteration number showing local minimum.

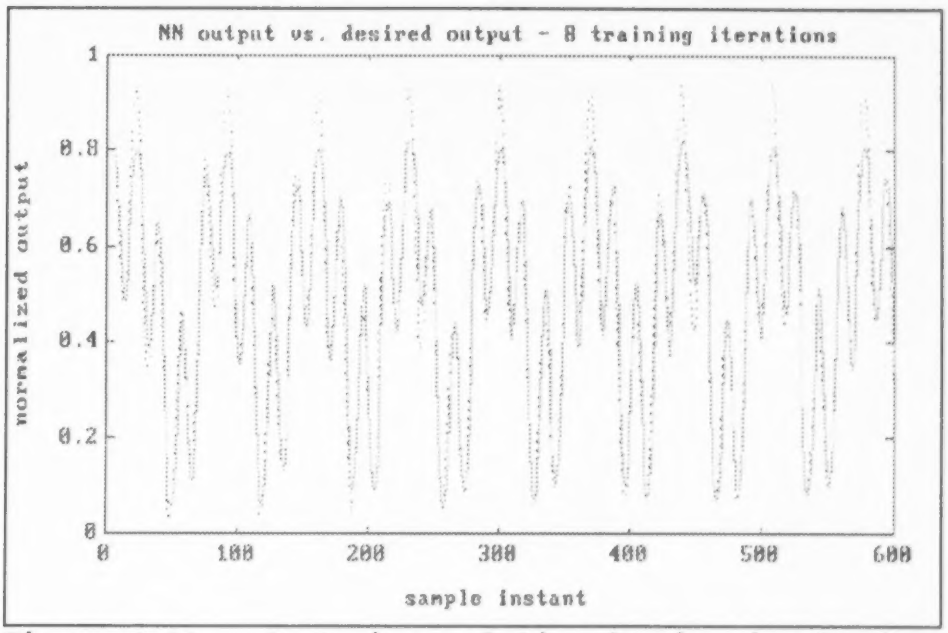

Figure 3.11. Comparison of time-domain signal after eight training iterations (solid=actual output of neural network, dotted=desired output).

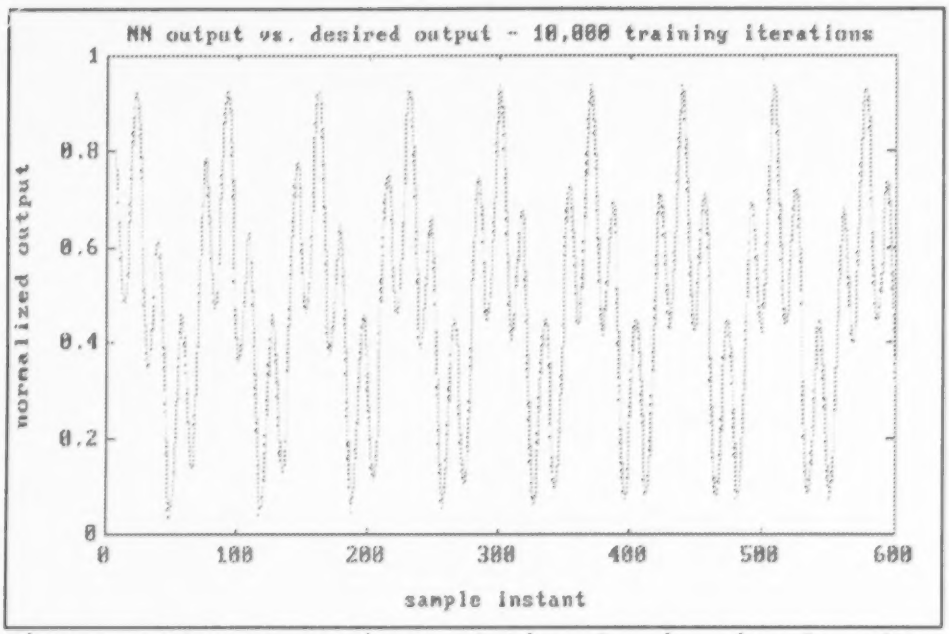

Figure 3.12. Comparison of time-domain signals after 10,000 training iterations (solid=actual output of neural network, dotted=desired output).

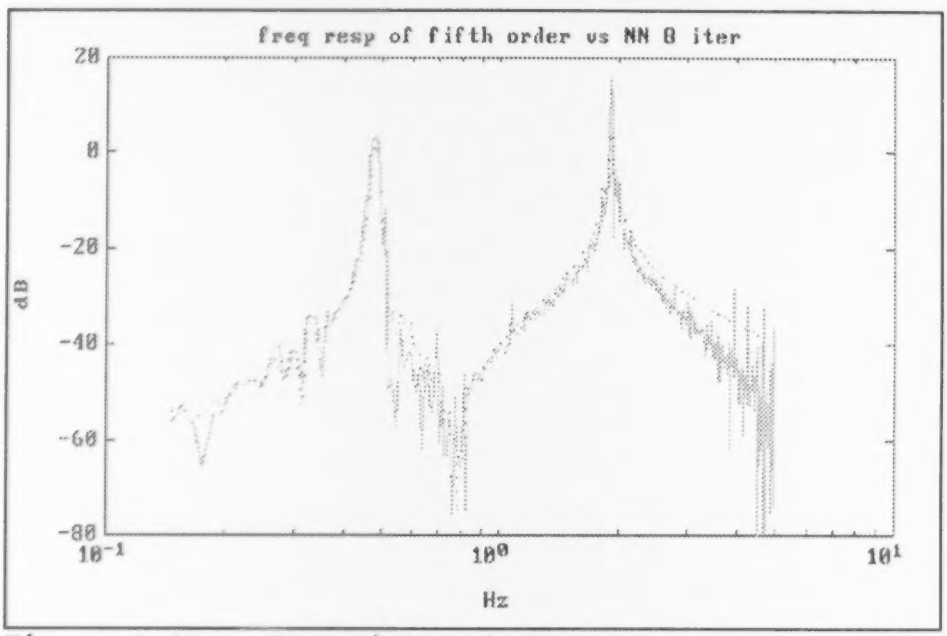

Figure 3.13. Comparison of frequency response after eight training iterations (solid=inverse neural network, dotted=fifth order system).

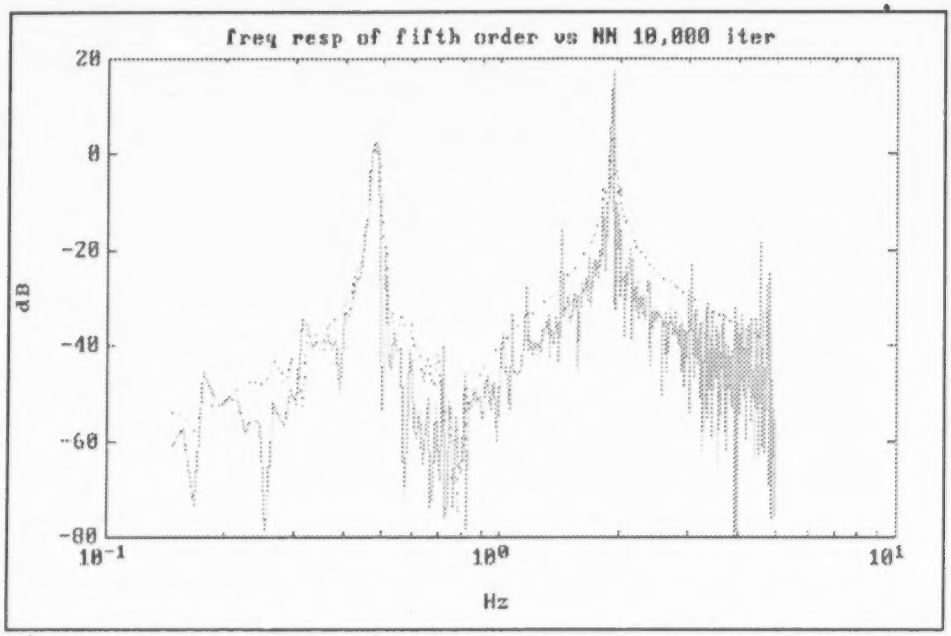

Figure 3.14. Comparison of frequency response after 10,000 training iterations (solid=inverse neural network, dotted=fifth order system).

Chapter 4

Performance Studies of the Trained Neural Network

A number of performance tests were done on the trained neural network to determine how well it could adapt to varying conditions while continuing to correctly identify the power system.

Particular attention was given to the neural network's ability to identify the system when training was stopped after only eight iterations. In order to present itself as a viable identification tool, the neural network must be able to capture the system's important characteristics in a short period of time. This necessitates an iteration constraint.

The first group of tests involve changes to the test system's operating conditions. Given that the test system was already a lightly-damped system, the damping of the local mode was decreased further while either increasing or decreasing the frequency. In addition, a third order transfer function was tested.

The second group of tests investigated the effect that network

configuration changes had on the neural networks ability to identify the fifth order transfer function. Training was done using a feedforward model, a feedback model, and a feedforward model while varying the number of nodes and node interconnections. Testing was then done by operating each neural network in either feedback mode or feedforward mode using different normalization procedures for the input.

The final test examined the ability of the neural network trained on a two-machine-infinite-bus system (fifth order transfer function), to correctly identify a single-machine-infinite-bus system (sixth order transfer function).

4.1 Changes in Operating Conditions

The following tests involved the identification of a fifth order power system transfer function with different transfer function coefficients from that used in the original training model. The frequency responses of the new power system and the neural network model are then compared.

The neural network was trained using the feedforward model. There were ten nodes in the input layer representing four delayed values of the power system input, one present value of power system output and

five delayed values of power system output. There were twenty nodes in the hidden layer, and one output node. The learning rate and momentum term were set to 0.25 and 0.9 respectively. The input training signal was the sinusoid of (3.7), with the input normalized between zero and one. The weights were frozen after eight iterations of training on the base case transfer function of (3.2).

In each case, the neural network was tested using a 512 point PRBS with pulse width of 0.2 seconds and a sampling rate of 40 Hz. The PRBS was passed through the new plant using the feedforward configuration and then through the fixed weight neural network.

A summary of the time-domain error for each configuration is included in table 4.1 at the end of this section. The least squares time-domain error of equation (2.6) was also calculated for each of the tests using the sinusoidal input of (3.7) and compared to the error of the base case fifth order transfer function.

4.1.1 Third order System

The first test involved setting the zero term equal to the interarea mode in the fifth order transfer function of (3.2), so that cancellation of the two results. The power system transfer function is now given by,

$$\frac{P_e}{V_{ref}} = \frac{25s}{(s+12.5)(s^2+0.1s+144)} \tag{4.1}$$

Figure 4.1 shows the frequency response of the third order system with a mode of oscillation at 1.9 Hz. The frequency response of the inverse neural network clearly shows that the neural network has captured the single oscillatory mode. Equally important is the fact that the magnitude of the plots are equal. Again, the dB per decade drop-off after the mode of oscillation is greater for the inverse neural network.

In the time-domain, the sum of the error squared has increased from 0.406 to 0.598 over all 600 patterns.

4.1.2 Decreased Damping and Reduced Frequency of Local Mode

The second test involved halving the frequency of the local mode and reducing the overall damping by a factor of five. This results in the following fifth order transfer function,

$$\frac{P_e}{V_{ref}} = \frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+9)(s^2+0.01s+36)} \tag{4.2}$$

Figure 4.2 shows the frequency response of the fifth order system with modes of oscillation at 0.48 Hz and 0.95 Hz. Figure 4.2 also shows the frequency response of the inverse neural network. Both oscillatory modes of the system have been identified. Again, the frequency response of the fifth order transfer function and the inverse neural network have the same magnitude plots. The neural network's dB roll-off of the local mode is somewhat steeper than that of the fifth order system.

The time-domain least-squared error has increased considerably more from 0.406 in the test system to 1.64 for this test.

4.1.3 Decreased Damping and Increased Frequency of Local Mode

The final test involved doubling the frequency of the local mode while reducing the damping by a factor of twenty. The following transfer function results,

$$\frac{P_e}{V_{ref}} = \frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+9)(s^2+0.01s+576)} \qquad (4.3)$$

Referring to figures 4.3, the system's two modes of oscillation can

be seen at 0.48 Hz and 3.8 Hz. The frequency response of the inverse neural network also shows both the local and interarea modes correctly identified. Along with the higher frequency of the local mode, aliasing problems are more evident. The use of a low pass filter would minimize the effects of aliasing due to the high frequency components.

The time-domain error has increased from 0.406 during training to 1.46 for the final test.

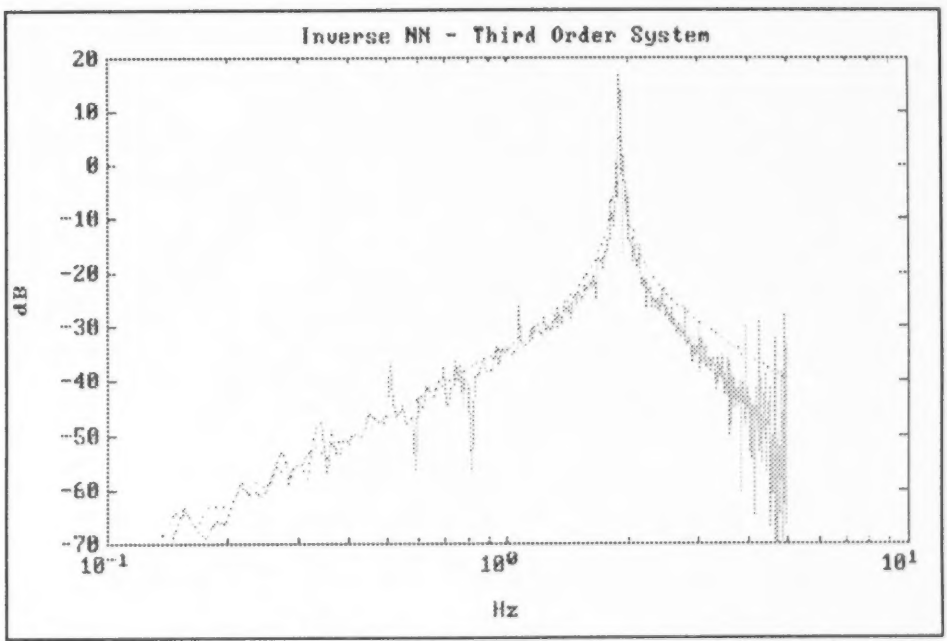

Figure 4.1. Comparison of frequency response of inverse neural network and third order test system (solid=neural network, dotted=third order model).

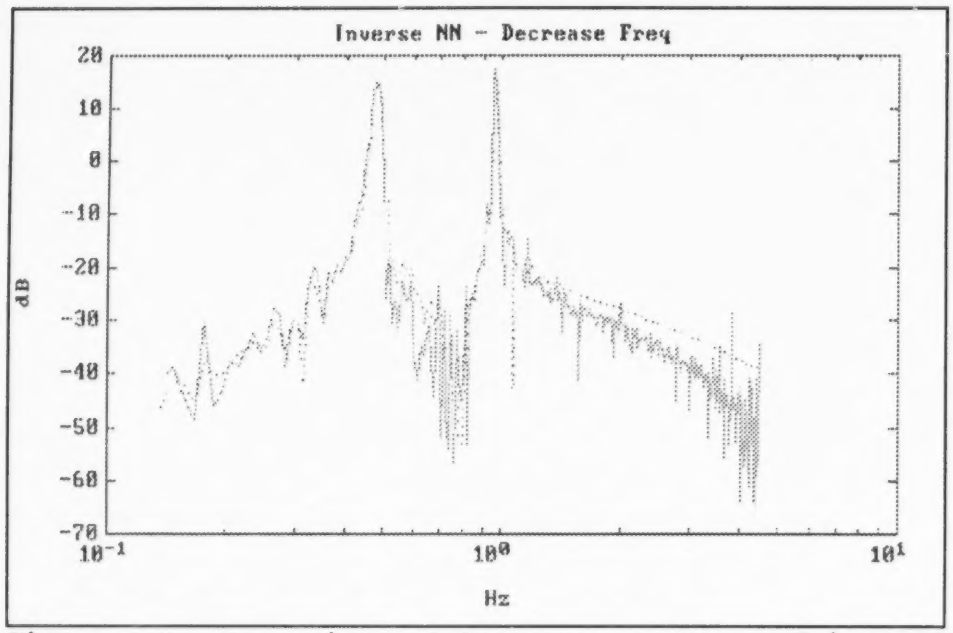

Figure 4.2. Comparison of frequency response of inverse neural network and fifth order test system with decreased damping and frequency (solid=neural network, dotted=fifth order model).

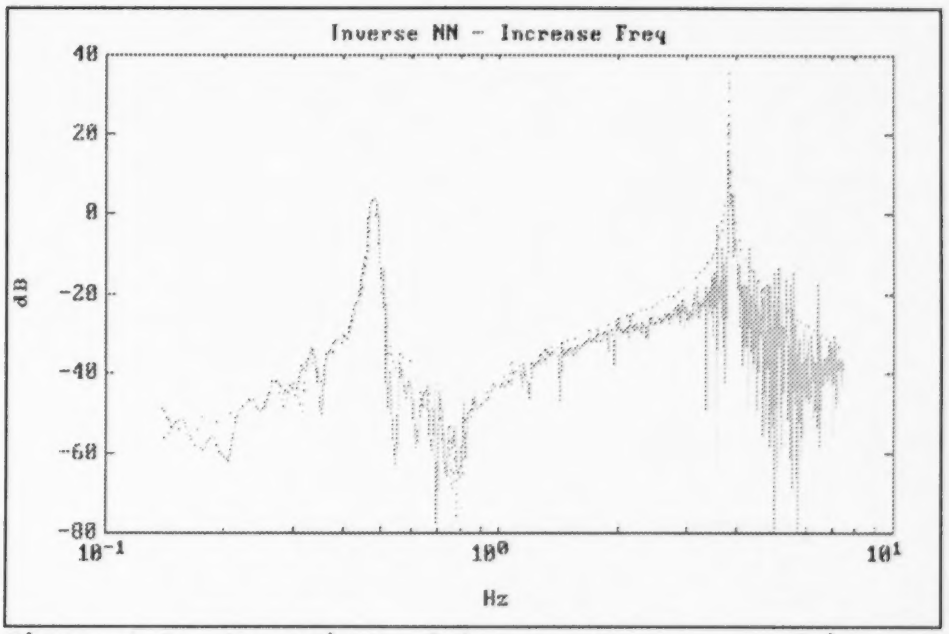

Figure 4.3. Comparison of frequency response of inverse neural network and fifth order test system with decreased damping and increased frequency (solid=neural network, dotted=fifth order model).

Transfer Function	Error $= \sum_p E_p$
$\frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+9)(s^2+0.1s+144)}$	0.406
$\frac{25s}{(s+12.5)(s^2+0.1s+144)}$	0.598
$\frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+9)(s^2+0.01s+36)}$	1.64
$\frac{25s(s^2+0.1s+25)}{(s+12.5)(s^2+0.1s+9)(s^2+0.01s+576)}$	1.46

Table 4.1. Least squares time-domain error for sinusoidal input.

4.2 Changes to Neural Network Configuration

The following tests involve changes to either the neural network training configuration, the testing configuration, or both. In all cases, the training input was the sinusoid described by (3.7) and the learning rate and momentum term were 0.25 and 0.9 respectively. Each neural network was trained for eight iterations at which time the weights were frozen for testing. All of the various architectures were tested on the base case fifth order transfer function described by (3.1). In all cases, a 512 point PRBS input signal with a minimum pulse width of 0.2 seconds and 40 Hz sampling were used to obtain the frequency response.

The training and testing included various combinations of input normalized in the range zero to one, input normalized in the range negative one to positive one, and testing in both feedforward mode and feedback mode. Table 4.2 at the end of this section is a summary of the various tests.

4.2.1 Feedforward Training

A neural network was trained in feedforward mode with ten input nodes, twenty hidden layer nodes, and one output node. The sinusoidal input was normalized from zero to one for both training and testing, and

then the y_p inputs were normalized from negative one to positive one for both training and testing. Testing was then done by using the feedforward mode and then by using the feedback mode for each case.

Referring to figures 4.6 and 4.7, the neural network was able to identify both oscillatory modes when tested using the feedforward mode, regardless of the normalization factor used. The magnitude of the peaks is comparable, although the neural network has a slightly lower dB roll-off on the local mode. The negative one to positive one input range produced the same neural network frequency response as the zero to one input range except a bias has been added.

As seen in figures 4.8 and 4.9, the neural network was unable to identify the system oscillatory modes when tested in feedback mode.

4.2.2 Feedback Training

The neural network was trained using the feedback configuration with ten input nodes, twenty hidden layer nodes, and one output node. Although there was no guarantee of convergence, of interest was whether the neural network could capture any of the test system's important characteristics. The same procedure as given in section 4.2.1 was followed for training except the output of the neural network becomes a delayed input for training as shown in figure (3.5).

The time-domain training errors are much higher for this test: 10.797 and 13.618 for input range zero to one and range negative one to positive one respectively, as compared to 0.406 and 0.793 for the test case. In spite of this, it can be seen from figures 4.10 and 4.11 that the neural network has identified both oscillatory modes when tested in the feedforward mode, even though the frequency response is not clearly defined. However, the slopes and the magnitudes of the oscillatory modes of the neural network frequency response are not the same as the fifth order system. The negative one to positive one input range again causes a bias in the frequency response of the neural network.

As shown in figures 4.12 and 4.13, when tested in the feedback mode, the neural network is unable to identify either the local or interarea mode of oscillation.

4.2.3 Number of Nodes and Node Interconnections

In this section, various architectures with fewer nodes and node interconnections are investigated. The motivation for doing so is to reduce the computational effort necessary for identification, resulting in less time needed. Although the accuracy of the model may be reduced as a result, the purpose of the identification scheme presented here is for subsequent controller design rather than exact modelling.

The first node reduction test was done by reducing the number of inputs into the neural network. From the functional relationship given in equation (3.6), the output of the neural network is a function of the power system's past inputs and past outputs. Since the past inputs only determine the numerator term of the power system transfer function, only the two most recent input terms were used for training. This results in the following functional relationship,

$$u_n(k-1) = g\left[u_p(k-2),u_p(k-3),\ y_p(k),y_p(k-1),\ldots,y_p(k-5)\right] \tag{4.4}$$

In order to account for all the inputs defined by (4.4), eight inputs are necessary for the neural network. In light of the reduced number of input nodes, it was arbitrarily decided to reduce the hidden layer nodes from twenty to sixteen. All other training and testing parameters remained as before.

As can be seen from figures 4.14 and 4.15, the neural network clearly captures both oscillatory modes, regardless of the normalized input range when tested in feedforward mode. When tested using the input range of zero to positive one, the only difference between the neural network's frequency response and that of the fifth order system

is the slightly higher roll-off of the local mode identified by the neural network. The negative one to positive one input range has again resulted in a positive bias on the neural network's frequency response. Again, the neural network is unable to identify the fifth order system when tested in feedback mode, as seen in figures 4.16 and 4.17.

The number of weight and bias connections has been reduced from 241 in the base case test system to 161 for the eight input node/sixteen hidden node case. This resulted in a 29% reduction in computational time per iteration, as compared to the iteration time of the base case test system.

The second test was to reduce the weight interconnections by separating the coupling between the neural network input terms. This was accomplished by removing the interconnected weights between the first layer and the hidden layer. This resulted in the four past u_p terms feeding from the input layer into eight hidden layer nodes, and the six past y_p terms feeding from the input layer into twelve hidden layer nodes. As can be seen from figure 4.4, the number of interconnections (weights) have been reduced considerably. This results in a decrease in computational time since fewer weights have to be updated on each iteration.

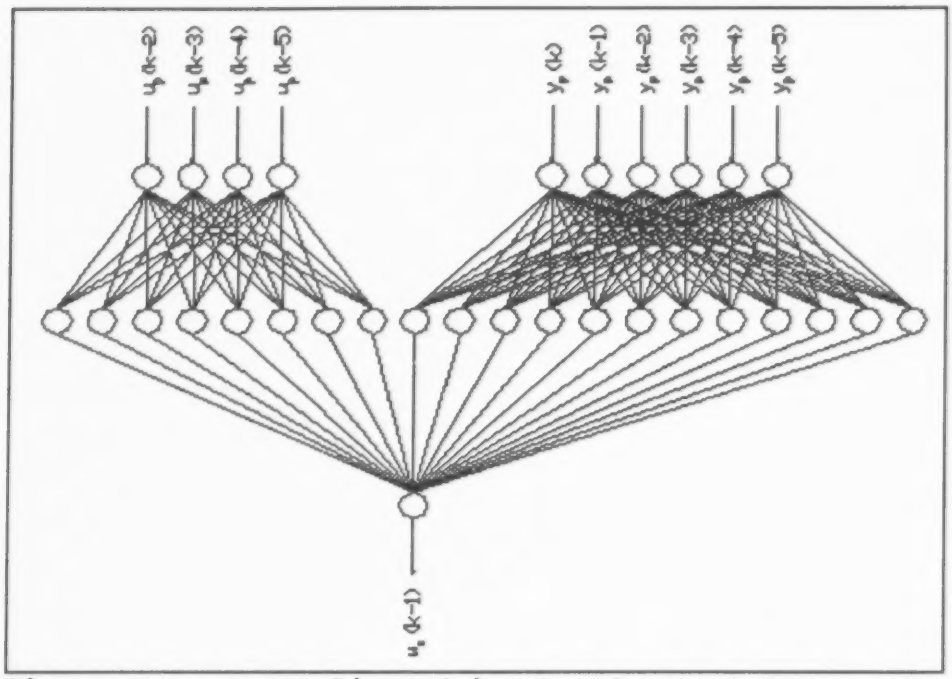

Figure 4.4. Uncoupling of input and output terms.

The same training and testing procedure was again followed. Referring to figures 4.18 and 4.19, the neural network is able to identify both the local and interarea modes when tested in feedforward mode, regardless of the normalized input. The zero to positive one input range results in a frequency response that closely matches the fifth order system in magnitude, except for the slopes of the local mode, which are slightly different. The negative one to positive one input range produces

the same frequency response with the addition of a positive bias for the neural network plot.

As seen in figures 4.20 and 4.21, the neural network is again unable to identify the fifth order test system when tested in feedback mode, regardless of the range of the input excitation.

The number of weights has now been reduced from 241 in the base case test system to 145 in this test. This resulted in a decrease in computational time of 37% over the base case.

The third reduction test is a combination of tests one and two where both the number of past plant input terms used as input to the neural network has been reduced to two, and the coupling between inputs and outputs has been separated. This resulted in two u_p terms feeding into four hidden layer nodes, and six y_p terms feeding into twelve hidden layer nodes, with all 16 nodes joining at the output node of the neural network. This new architecture is shown in figure 4.5.

The same training and testing procedures were followed as before. As can be seen from figures 4.22, and 4.23, the neural network is able to identify both oscillatory modes when tested in feedforward mode. The input normalized in the range zero to positive one results in a fairly close match in the magnitudes of the plots. The roll-off of both the

interarea mode and the local mode are slightly lower for the neural network, but the magnitude of the peaks are the same. Again, the input range negative one to positive one results in a positive bias on the frequency response plot of the neural network. Although the roll-offs of the oscillatory modes more closely match, the magnitude of the peaks does not.

As can be seen in figures 4.24 and 4.25, although the frequency response of the neural network has the general shape of the fifth order system's frequency response, it is unable to identify system clearly, regardless of the input range used.

The number of weights has decreased from 241 in the base case test system to only 113 in this test, resulting in a 51% reduction in computing time per iteration.

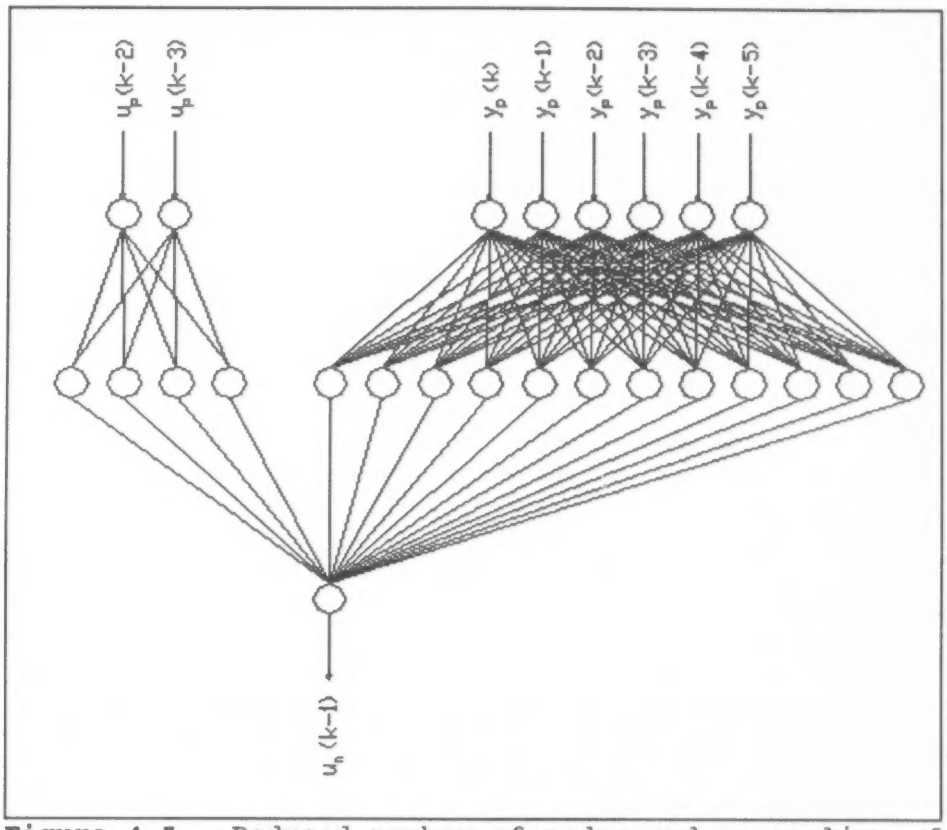

Figure 4.5. Reduced number of nodes and uncoupling of input and output terms.

Test 1: Feedforward Training

Feedforward Testing

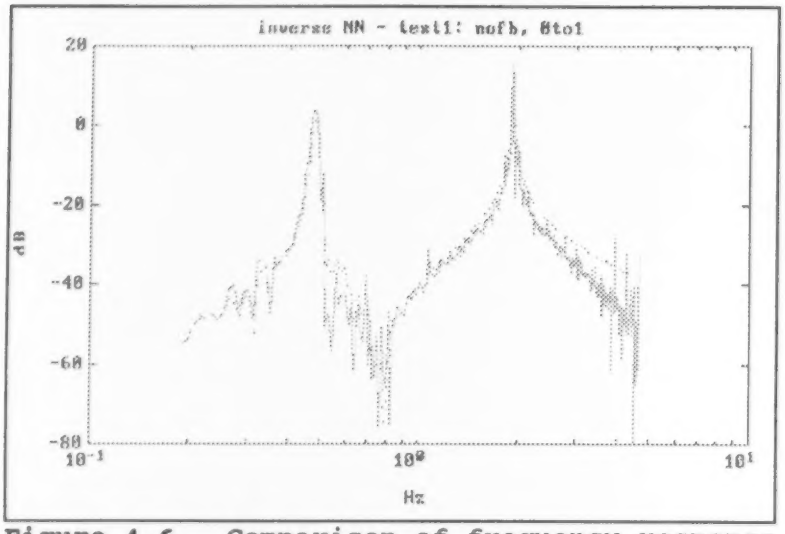

Figure 4.6. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order system).

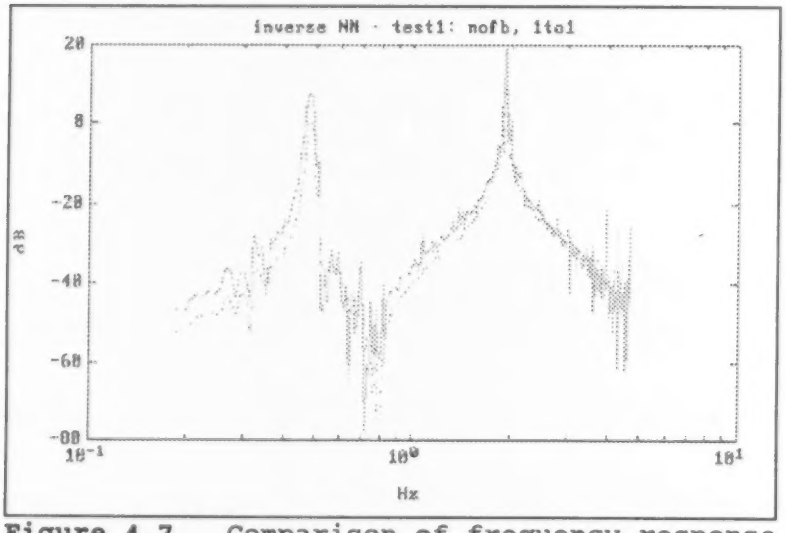

Figure 4.7. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order model).

Test 1: Feedforward Training

Feedback Testing

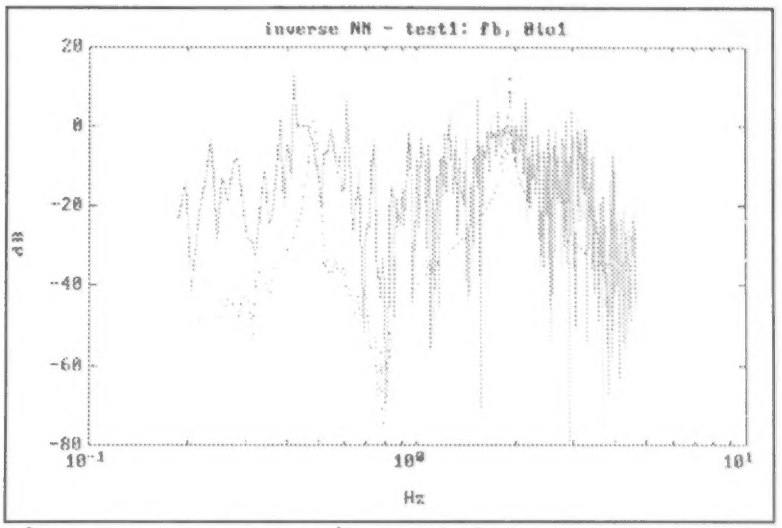

Figure 4.8. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

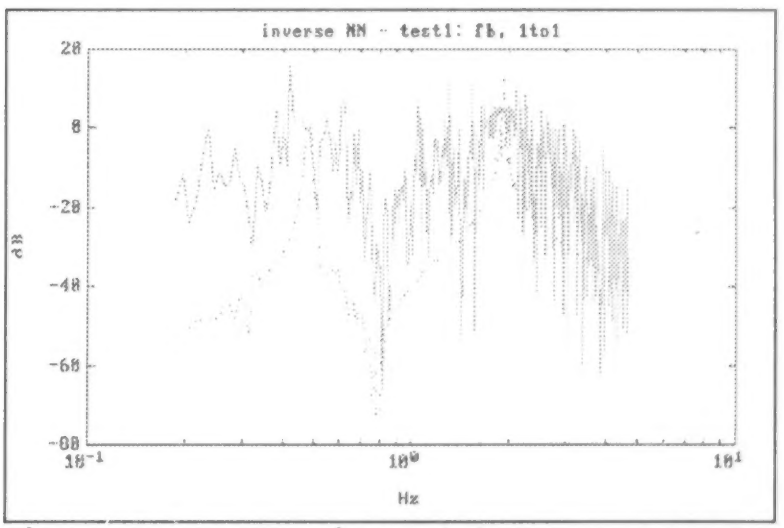

Figure 4.9. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 2: Feedback Training

Feedforward Testing

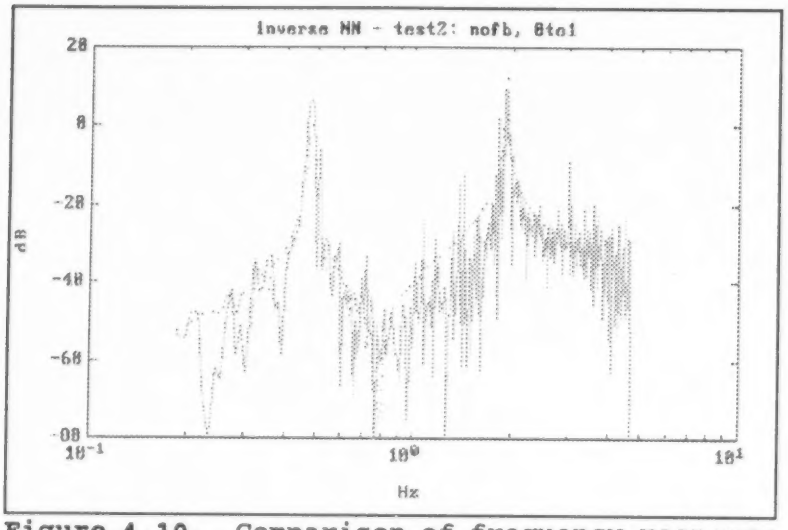

Figure 4.10. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

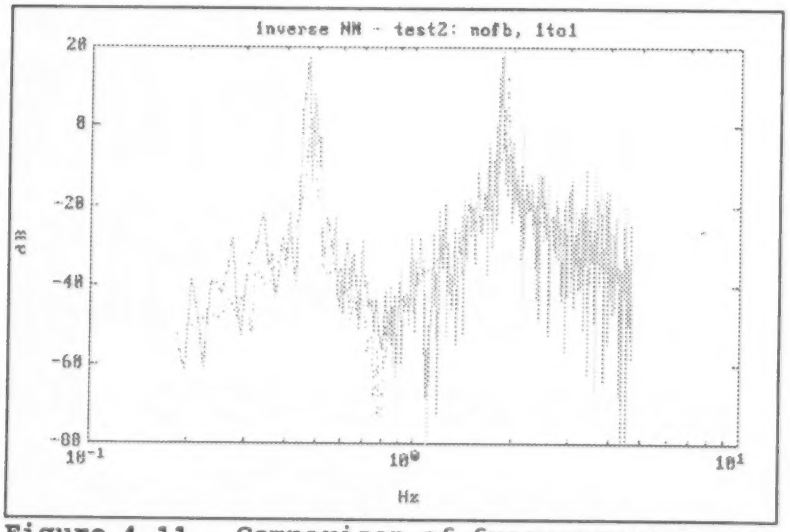

Figure 4.11. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 2: Feedback Training

Feedback Testing

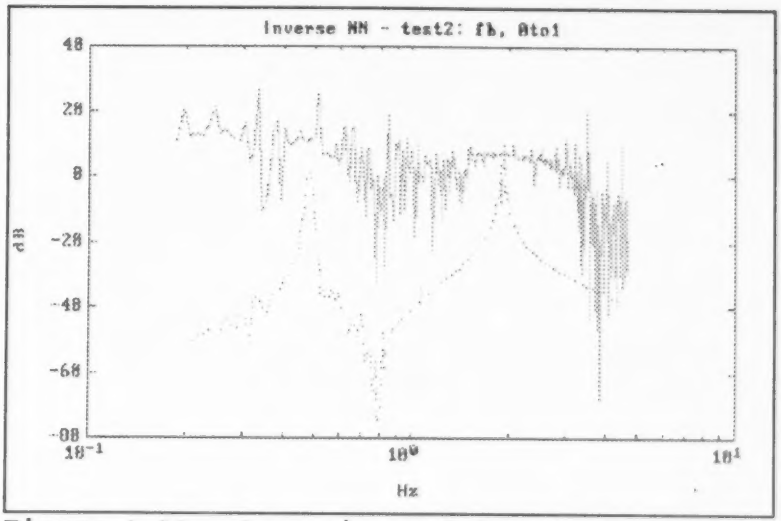

Figure 4.12. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

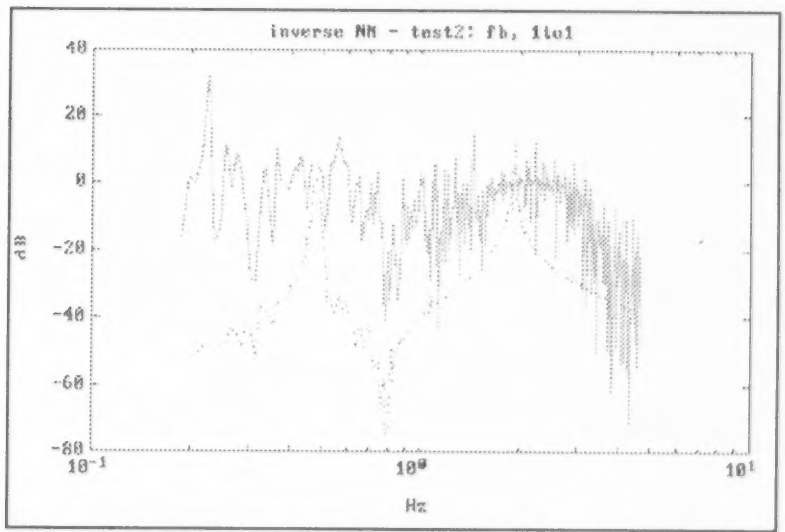

Figure 4.13. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 3: Node Reduction Feedforward Training

Feedforward Testing

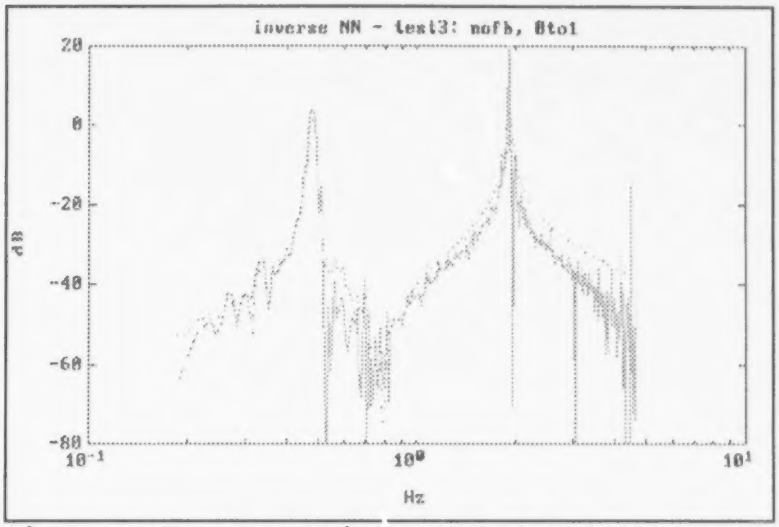

Figure 4.14. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

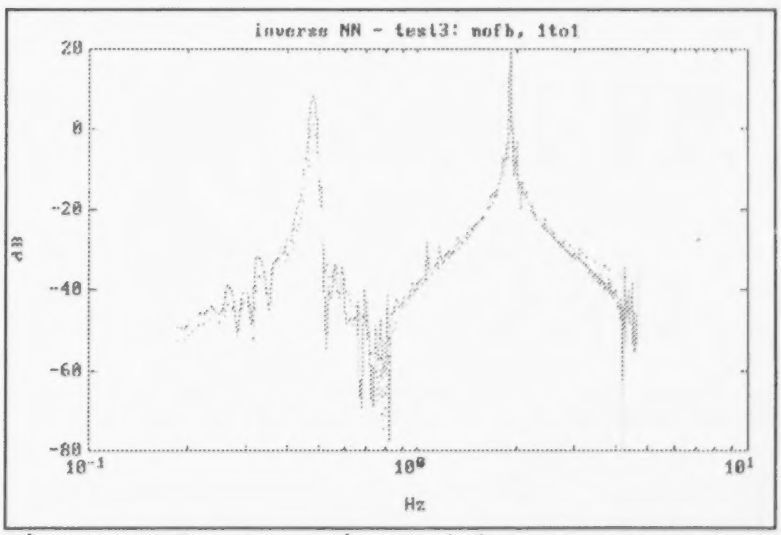

Figure 4.15. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 3: Node Reduction Feedforward Training

Feedback Testing

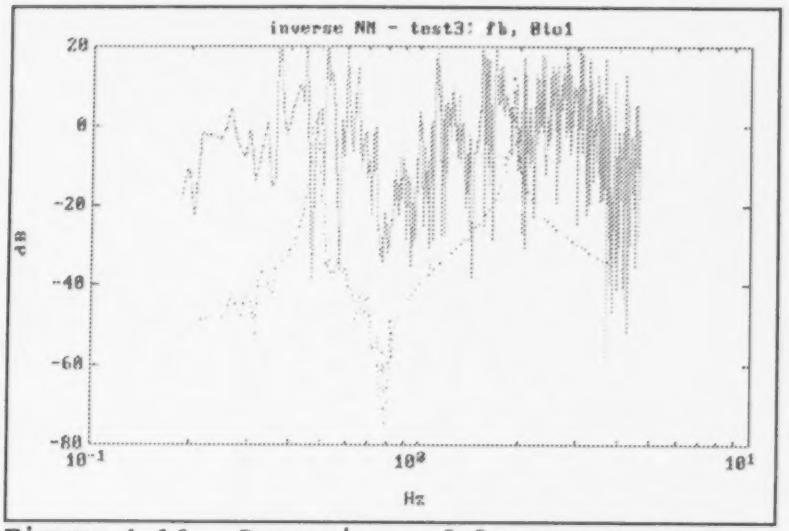

Figure 4.16. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

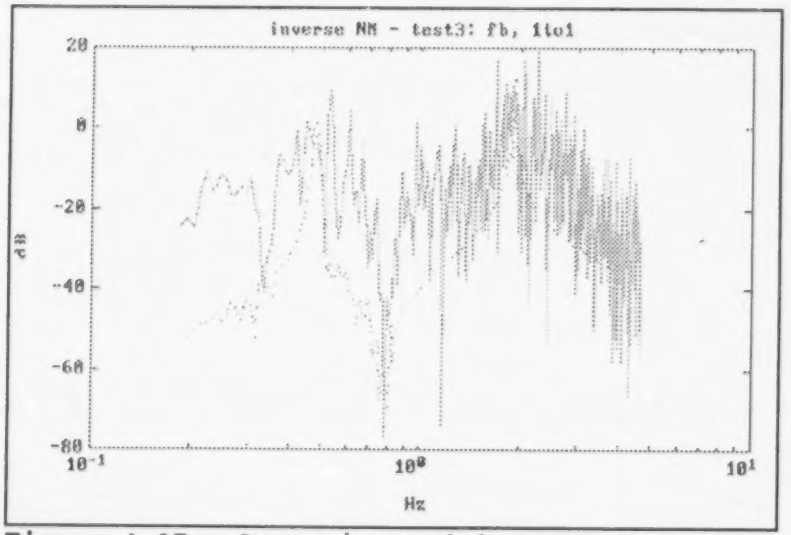

Figure 4.17. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 4: Uncoupling Feedforward Training

Feedforward Testing

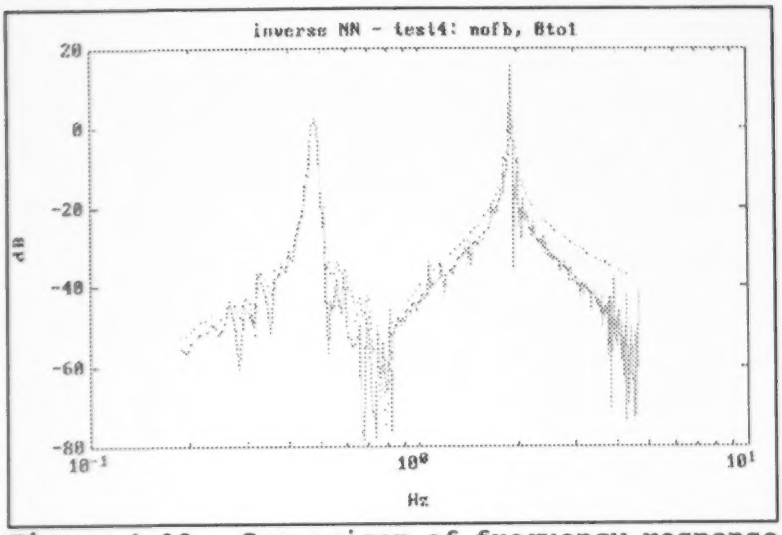

Figure 4.18. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

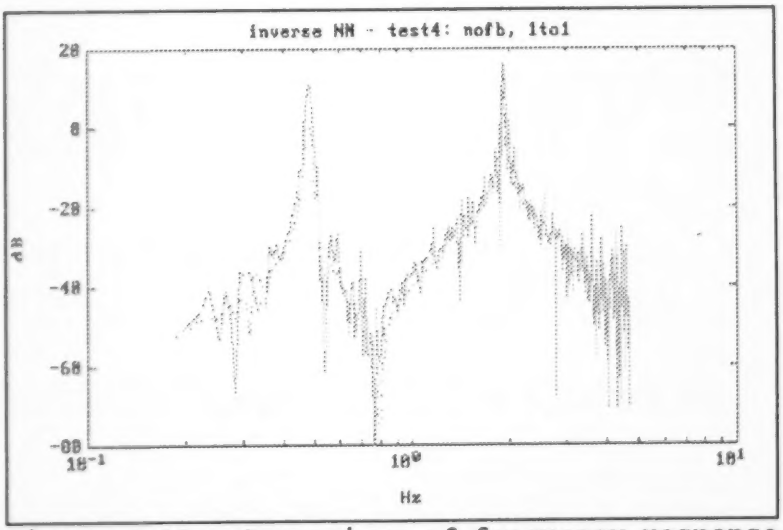

Figure 4.19. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 4: Uncoupling Feedforward Training

Feedback Testing

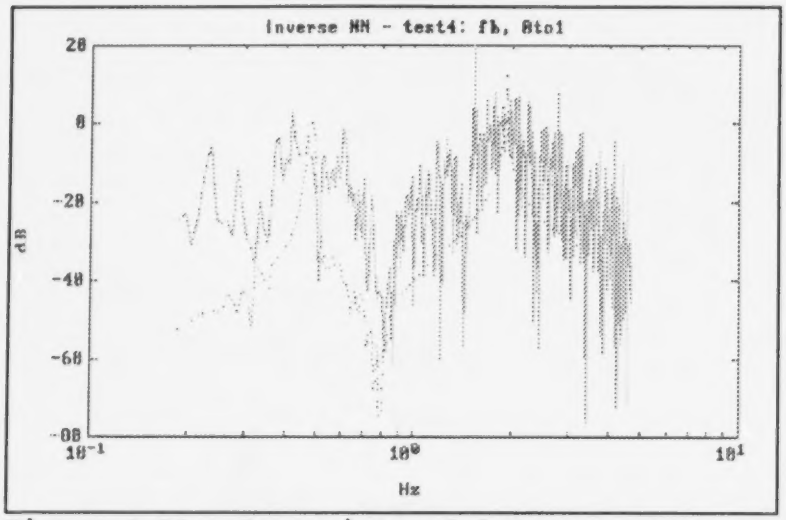

Figure 4.20. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

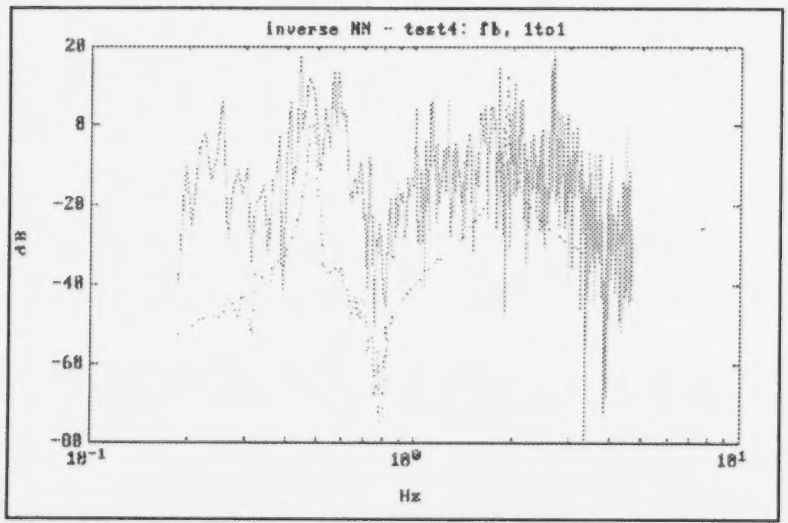

Figure 4.21. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 5: Node Reduction and Uncoupling Feedforward Training

Feedforward Testing

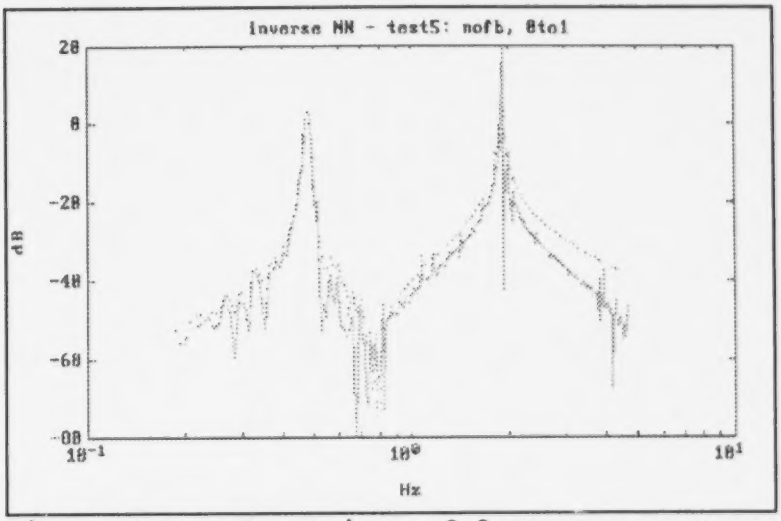

Figure 4.22. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

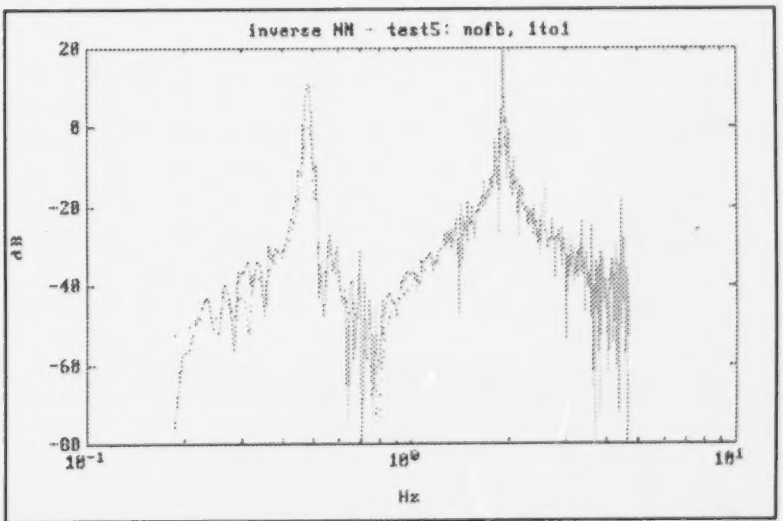

Figure 4.23. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Test 5: Node Reduction and Uncoupling Feedforward Training

Feedback Testing

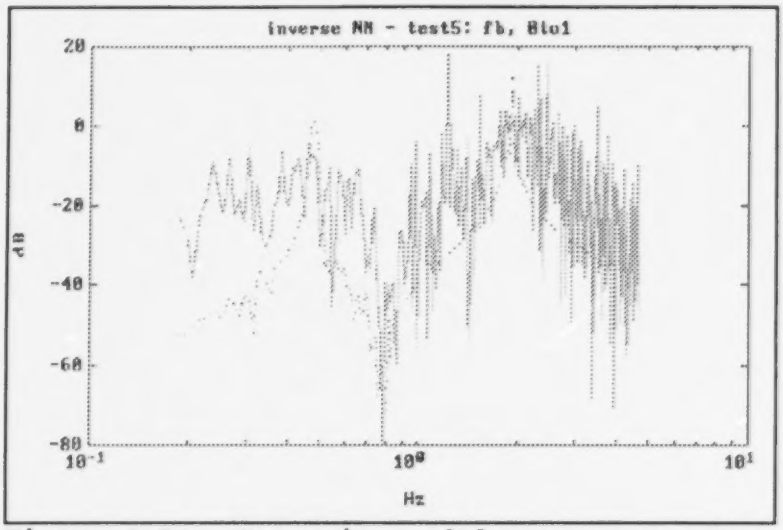

Figure 4.24. Comparison of frequency response of inverse neural network and fifth order test system - input range= 0 to +1 (solid=neural network, dotted=fifth order model).

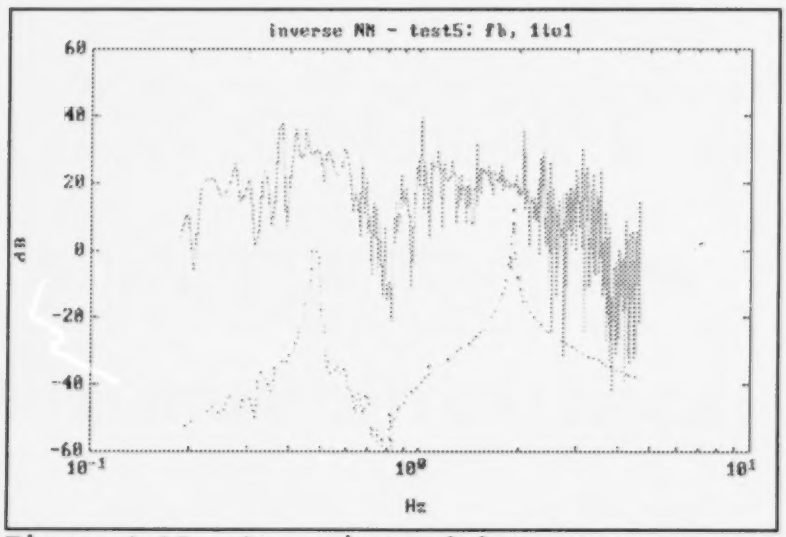

Figure 4.25. Comparison of frequency response of inverse neural network and fifth order test system - input range= -1 to +1 (solid=neural network, dotted=fifth order system).

Training = 8 iterations				Number of Network Weights	y_p Input Range	Train -ing Error
Test	**Input Nodes**	**Hidden Nodes**	**Out-put Node**			
Test 1					0 to +1	0.406
Feedforward Training	10	20	1	241	-1 to +1	0.793
Test 2					0 to +1	10.797
Feedback Training	10	20	1	241	-1 to +1	13.618
Test 3					0 to +1	2.062
Node Reduction	8	16	1	161	-1 to +1	0.933
Test 4					0 to +1	1.026
Uncoupling	4 6	8 12	1	145	-1 to +1	2.429
Test 5					0 to +1	0.418
Reduction & Uncoupling	2 6	4 12	1	113	-1 to +1	5.274

Table 4.2. Summary of changes to neural network configuration for feedforward testing.

4.3 Test on Single-Machine-Infinite-Bus Model

As a final test, the neural network trained on the fifth order transfer function from the two-machine-infinite-bus system, was tested on a sixth order transfer function derived from a single-machine-infinite-bus system (SMIB). The SMIB model was taken from [16], and the state space model technique was used to derive the transfer function as,

$$\frac{P_e}{V_{ref}} = \frac{334s(s+12.6)(s+6.24)(s+1.48)}{(s+17.9)(s+11.6)(s+8.5)(s+2.5)(s^2+0.63s+228)} \tag{4.5}$$

A 128 point PRBS with a minimum pulse width of 0.2 seconds and 40 Hz sampling was used as the excitation input with FFT techniques used to obtain the frequency response. The fixed neural network weights were used from the base case fifth order transfer function model trained for eight iterations in feedforward configuration. Testing was also done in feedforward mode with the results displayed in figure 4.26. The neural network has clearly identified the single oscillatory mode of the SMIB at 2.4 Hz. The magnitude of the peak is slightly higher for the frequency response of the neural network. The dB per decade roll-off after the mode of oscillation is the same, however, the frequency response of the neural network has approximately a 3 dB bias evident

before the oscillatory mode.

Test on Single-Machine-Infinite-Bus System

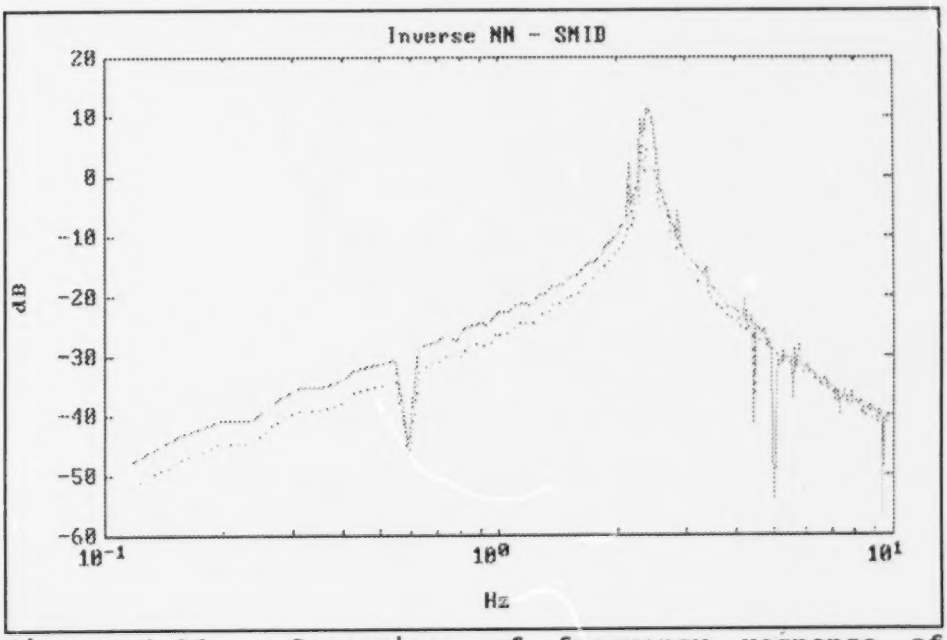

Figure 4.26. Comparison of frequency response of inverse neural network and single-machine-infinite-bus system (solid=neural network, dotted=SMIB).

Chapter 5

Summary and Conclusions

The main emphasis of this thesis is the development of a neural network for measuring the transfer function of a power system. In order to provide a viable alternative identification scheme, the neural network must quickly and accurately provide a model indicative of normal operating conditions without upsetting the system.

A fifth order transfer function with lightly-damped complex poles was chosen as the test system due to the difficulties associated with identifying two oscillatory modes that are close in frequency but differ by more than 10 dB in the amplitude of each mode. The frequency response of the simulated power system served as a standard of comparison for the frequency response of the trained neural network.

Typically, research that utilizes neural networks for the identification and control of power systems, is focused on the control aspect, and thus the identification procedure consists of thousands of iterations to minimize the time-domain error.

This research emphasizes a quick, accurate transfer function identification scheme with comparisons done in the frequency domain between the simulated system and the neural network.

Pseudo-Random-Binary-Sequence excitation and Fast Fourier Transform analysis were used to obtain the frequency response models. In order to provide a higher energy level to the modes of oscillation, pre-filtering of the excitation signal was used.

A static multilayered neural network was trained to identify the inverse dynamics of the fifth order transfer function. An examination of the sum of the squared error showed a local minimum at the eighth training iteration. Rapid convergence is a desirable quality of any identification scheme, and since the frequency response of the inverse neural network at the eighth iteration varied little from that at the ten thousandth iteration, eight training iterations was used as a benchmark for the simulation studies. It is recognized that a more exact model would require considerably more training time. However, the focus of this work was to provide a sufficiently accurate model for subsequent controller design.

The benchmark neural network provided good agreement with the fifth order transfer function model. Both modes of oscillation were correctly identified, both for the frequency and amplitude of each

oscillatory mode, and for the slopes of each mode.

The results of the simulation studies were presented in Chapter 4. The robustness of the neural network was investigated by simulating changes in operating conditions. A third order transfer function, and a fifth order transfer function with decreased damping and either increased or decreased frequency were tested utilizing the fixed weights from the benchmark neural network. Given that the weights of the neural network were fixed during testing, the sum of the squared error did increase, however, all frequency responses displayed reasonable accuracy in identifying both modes of oscillation and in the amplitude of the modes.

Also investigated were changes to the neural network architecture to see how well it could identify the fifth order transfer function while decreasing the computational time needed to do so. This was done by either decreasing the neural network inputs, the number of hidden layer nodes, or by separating the coupling between the neural network input terms. Although network architecture is system specific, computational times were decreased between 29% and 51% with respect to the time needed for the benchmark neural network, while retaining a reasonably accurate model of the system.

The effects of the normalization procedure were tested by

increasing the normalization of the input to the range negative one to positive one from the training range of zero to positive one. This resulted in a positive bias in the frequency response.

The differences between using a feedforward model and a feedback model for either training or testing was briefly investigated. Testing using the feedforward model provided fairly reasonable frequency responses in all cases, whereas the feedback model did not.

Finally, the benchmark neural network was tested on a sixth order transfer function that was based on a single-machine-infinite-bus model. Good agreement resulted from the comparison of their respective frequency responses.

In summary, the main contributions of this research were:

(1) The development of an alternative identification tool for power systems. Reasonable convergence was obtained in eight iterations for this system. Simulation studies showed the neural network to be robust to changes in operating conditions.

(2) Methods of reducing the computational time necessary for the identification procedure included both reducing the number of neural network inputs and separating the coupling between the inputs. Computational times were decreased by up to 51% over the benchmark neural network model.

5.1 Recommendations for Further Research

Simulations studies have suggested areas worthy of further study.

Test results suggest the feasibility of the neural network as a viable identification tool, however the effects of noise were not investigated. Practical considerations require that the effects of noise be studied before conclusions regarding the neural network's identification capabilities are made.

In addition, whether the neural network could successfully identify a multi-mode system containing machines with different inertia constants without the benefits of a pre-filtered excitation signal would be of interest.

Finally, efforts devoted towards a robust on-line identifier would improve the tracking ability of an adaptive controller based on neural network concepts.

Bibliography

(1) Balda, J.C., R.E. Fairbairn, R.G. Harley, J.L. Rodgerson and E. Eitelberg, "Measurement of Synchronous Machine Parameters by a Modified Frequency Response Method-Part II: Measured Results", *IEEE Transactions on Energy Conversion*, Vol. 2, No. 4, Dec. 1987, pp. 652-657.

(2) Balda, J.C., M.F. Hadingham, R.E. Fairbairn, R.G. Harley and E. Eitelberg, "Measurement of Synchronous Machine Parameters by a Modified Frequency Response Method-Part I: Theory", *IEEE Transactions on Energy Conversion*, Vol. 2, No. 4, Dec. 1987, pp. 646-651.

(3) Bendat, Julius S. and Allan G. Piersol, *Random Data: Analysis and Measurement Procedures*, Wiley-Interscience, New York, 1971.

(4) Beya, K., R. Pintelon, J. Schoukens, P. Lataire, P. Guillaume, B. Mpanda-Mabwe and M. Delhaye, "Identification of Synchronous Machines Parameters Using Broadband Excitations", *IEEE Transactions on Energy Conversion*, Vol. 9, No. 2, June 1994, pp. 270-280.

(5) Boje, E.S., J.C. Balda, R.G. Harley and R.C. Beck, "Time-Domain Identification of Synchronous Machine Parameters from Simple Standstill Tests", *IEEE Transactions on Energy Conversion*, Vol. 5, No. 1, Mar. 1990, pp. 164-175.

(6) Bollinger, K.E., H.S. Khalil, L.C.C. Li and W.E. Norum, "A Method For On-line Identification of Power System Model Parameters in the Presence of Noise", *IEEE Transactions on Power Apparatus and Systems*, Vol. 101, No. 9, Sep. 1982, pp. 3105-3111.

(7) Bollinger, K.E., H.S. Khalil and W.E. Norum, "Practical Identification Techniques for Excitation Systems", *CEA Spring Meeting*, Vol. 19, Part 5, Mar. 1980, pp. 80-SP-170/1-15.

(8) Bollinger, K.E., A.K. Laha and R.A. Winsor, "System Models from Transient Stability Programs", In *IEEE PICA, New Orleans, Proceedings*, July 1976, pp. TP X11-6 330-334.

(9) Bollinger, K.E. and W. Eric Norum, "Time Series Identification of Interarea and Local Generator Resonant Modes", *IEEE/PES Winter Meeting*, Jan. 1994, pp. 94 WM 180-0 PWRS/1-6.

(10) Bollinger, K.E., R. Winsor and D. Cotcher, "Power System Identification Using Noise Signals", *IEEE/PES Summer Meeting*, July 1976, pp. A76 339-2/1-7.

(11) Chen, Fu-Chuang. "Back-Propagation Neural Networks For Nonlinear Self-Tuning Adaptive Control", *IEEE Control Systems Magazine*, Apr. 1990, pp. 44-48.

(12) Chow, Mo-yuen and Robert J. Thomas, "Neural Network Synchronous Machine Modeling", *International Symposium on Circuits and Systems*, Vol. 1, 1989, pp. 495-498.

(13) deMello, F.P. and L.N. Hannett, "Determination of Synchronous Machine Electrical Characteristics By Test", *IEEE Transactions on Power Apparatus and Systems*, Vol. 102, No. 12, Dec. 1983, pp. 3810-3815.

(14) de Villiers, Jacques and Etienne Barnard, "Backpropagation Neural Nets With One and Two Hidden Layers", *IEEE Transactions on Neural Networks*, Vol. 4, No. 1, Jan. 1992, pp. 136-141.

(15) El-Sharkawi, M.A., A.A. El-Samahy and M.L. El-Sayed, "High Performance Drive of DC Brushless Motors Using Neural Network", *IEEE Transactions on Energy Conversion*, Vol. 9, No. 2, June 1994, pp. 317-322.

(16) Gu, Wenyan and K.E. Bollinger, "A Decoupler Design Method For Synchronous-Machine-Infinite-Bus System", *IEEE Transactions on Energy Conversion*, Vol. 4, No. 1, Mar. 1989, pp. 54-61.

(17) Hobson, E. and G.N. Allen, "Effectiveness of Artificial Neural Networks for First Swing Stability Determination of Practical Systems", *IEEE Transactions on Power Systems*, Vol. 9, No. 2, May 1994, pp. 1062-1068.

(18) Hoskins, D.A. and J. Vagners, "A Neural Network Based Explicit Model Reference Adaptive Controller", *Proceedings of the 29th Conference on Decision and Control*, Vol. 3, Dec. 1990, pp. TA-12-1-9:40/1725-1729.

(19) Hsu, Yuan-Yih and Lin-Her Jeng, "Analysis of Torsional Oscillations Using An Artificial Neural Network", *IEEE Transactions on Energy Conversion*, Vol. 7, No. 4, Dec. 1992, pp. 684-690.

(20) Hunt, K.J., D. Sbarbaro, R. Zbikowski and P.J. Gawthrop "Neural Networks for Control Systems-A Survey", *Automatica*, Vol. 28, No. 6, 1992, pp. 1083-1112.

(21) Hush, Don R. and Bill G. Horne, "Progress in Supervised Neural Networks: What's New Since Lippman?", *IEEE Signal Processing Magazine*, Jan. 1993, pp. 8-39.

(22) Hush, Don R., Bill Horne and John M. Salas, "Error Surfaces for Multilayer Perceptrons", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 5, Sep/Oct. 1992, pp. 1152-1161.

(23) Lang, R.D., M.A. Hutchison and H. Yee, "Microprocessor-based Identification System Applied to Synchronous Generators With Voltage Regulators", *IEE Proceedings*, Vol. 130, Part C, No. 5, Sep. 1983, pp. 257-265.

(24) Lari-Najafi, H., Mohammed Nasiruddin and Tariq Samad, "Effect of Initial Weights on Back-Propagation and its Variations", *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, 1989, pp. 218-219.

(25) Le, L.X. and W.J. Wilson, "Synchronous Machine Parameter Identification: A Time Domain Approach", *IEEE Transactions on Energy Conversion*, Vol. 3, No. 2, June 1988, pp. 241-248.

(26) Ljung, L. "Choices of Prefilters and Forgetting Factors in Noise-Free Identification", *Proceedings of the 1989 American Control Conference*, Vol. 2, June 1989, pp. TP1-2:30/1409-1415.

(27) Manchur, G., David C. Lee, M.E. Coultes, J.D.A. Griffin and Wilfred Watson, "Generator Models Established by Frequency Response Tests on a 555 MVA Machine", *IEEE Transactions on Power Apparatus and Systems*, 1972, pp. 2077-2084.

(28) Marpaka, D.R., Michael H. Thursby and Seyed M. Aghili, "Artificial Neural Net Based Stability Study of Power Systems", *IEEE Proceedings of Southeastcon*, Vol. 1, Apr. 1991, pp. 234-238.

(29) McClelland, James L. and David E. Rumelhart, *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*, MIT Press, Cambridge, 1988.

(30) Narendra, Kumpati S. and Kannan Parthasarathy, "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, Mar. 1991, pp. 252-262.

(31) Narendra, Kumpati S. and Kannan Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, Mar. 1990, pp. 4-26.

(32) Nguyen, Derrick H. and Bernard Widrow, "Neural Networks For Self-Learning Control Systems", *IEEE Control Systems Magazine*, Apr. 1990, pp. 18-23.

(33) Norum, William Eric. *Multimode Damping of Synchronous Generator Transients Using Self-Tuning Control*, PhD thesis, University of Alberta, Edmonton, Alberta, 1991.

(34) Psaltis, D., Athanasios Sideris and Alan A. Yamamura, "A Multilayered Neural Network Controller", *IEEE Control Systems Magazine*, Apr. 1988, pp. 17-20.

(35) Rumelhart, D.E., G.E. Hinton and R.J. Williams, "Learning Internal Representations By Error Propagation", *Parallel Distributed Processing, Volume 1: Foundations*, Ch. 8, MIT Press, 1986, pp. 318-362.

(36) Saerens, M. and A. Soquet, "Neural Controller Based on Back-Propagation Algorithm", *IEE Proceedings*, Vol. 138, Part F, No. 1, Feb. 1991, pp. 55-62.

(37) Smith, J.R., J.F. Hauer and D.J. Trudnowski, "Transfer Function Identification in Power System Applications", *IEEE Transactions on Power Systems*, Vol. 8, No. 3, Aug. 1993, pp. 1282-1290.

(38) Sriharan, S. and Koh Wee Hiong, "Synchronous Machine Modelling By Standstill Frequency Response Tests", *IEEE Transactions on Energy Conversion*, Vol. 2, No. 2, June 1987, pp. 239-245.

(39) Weerasooriya, Siri and M.A. El-Sharkawi, "Identification and Control of a DC Motor Using Back-Propagation Neural Networks", *IEEE Transactions on Energy Conversion*, Vol. 6, No. 4, Dec. 1991, pp. 663-669.

(40) Weerasooriya, Siri and Mohamed A. El-Sharkawi, "Laboratory Implementation of a Neural Network Trajectory Controller for a DC Motor", *IEEE Transactions on Energy Conversion*, Vol. 8, No. 1, Mar. 1993, pp. 107-113.

(41) Widrow, Bernard and Michel Bilello, "Adaptive Inverse Control", *Proceedings of the 1993 International Symposium on Intelligent Control*, Aug. 1993, pp. TA1-8:30/1-6.

(42) Zhang, Y., G.P. Chen, O.P. Malik and G.S. Hope, "An Artificial Neural Network Based Adaptive Power System Stabilizer", *IEEE Transactions on Energy Conversion*, Vol. 8, No. 1, Mar. 1993, pp. 71-77.